Credit Card Default Prediction: A Model Comparison

Student Numbers: 24059844, 24055736, 24182406, 24202682, 19074278

Abstract

Credit card delinquencies and balances have reached unprecedented levels, highlighting the growing need for reliable default prediction methods. This study compares six classification models – random and naive baselines, logistic regression, k-nearest neighbours, multi-layer perceptron, random forest, and XGBoost – on a well-known cross-sectional Taiwanese dataset. The data undergoes cleaning, one-hot encoding, and robust scaling, with the synthetic minority over-sampling technique (SMOTE) applied to mitigate severe class imbalance (22% default vs. 78% non-default). Performance is assessed via an 80/20 train-test split and 10-fold cross-validation, using balanced accuracy, weighted F_1, AUC-ROC, and Matthews correlation coefficient. Results show that logistic regression, random forest, and XGBoost achieve the highest predictive power and generalisability, with logistic regression especially benefiting from SMOTE. KNN and MLP exhibit larger performance drops under oversampling. Overall, the findings emphasise the importance of robust preprocessing and model selection for credit default prediction in imbalanced contexts.

1 Introduction

As of December 2024, credit card balances in the US have reached a record high total of $1.21 trillion [Federal Reserve Bank of New York, 2025]. In addition, the delinquency rate on credit card loans in the US across all commercial banks is currently 3.08%; the highest level since the aftermath of the 2008 credit crisis [Federal Reserve Bank of St. Louis, 2025]. Because the delinquency rate is an indicator of future write-offs, this is a major concern for credit institutions and macro-prudential regulators. The economic impact of defaults on credit institutions cannot be overstated. The cost of credit card default in the first three quarters of the year rose 50% year on year from 2023 to 2024, costing lenders $46bn in the first nine months of 2024 alone. This figure is expected to increase as concerns around consumers' financial health continue to grow [Financial Times, 2025].

The accurate prediction of creditors defaulting is a central and critical task for all credit institutions. The ultimate goal of credit card loan default prediction is to allow the credit institution to efficiently make decisions about which potential customers are credible. This task is now more critical than ever, given the growing delinquency rates, credit card balances, and concerns surrounding consumers' financial health. Failure to recognise bad loans could have dire economic consequences for individual institutions and the financial system as a whole, posing a huge systemic risk. As a result, this research is of great importance to credit institutions and macro-prudential regulators worldwide. This research aims to provide insights that will be of great value to these important economic agents.

As noted by Alam et al. [2020], classic statistical models tend to be unable to outperform modern machine learning techniques in credit default prediction due to the imbalanced nature of the data. As a result, we aim to provide an up-to-date overview of the most important traditional and cutting-edge machine learning methods capable of performing this classification task, analyse their efficacy, and ultimately identify the optimal model architecture. Namely, the six models we will compare are logistic regression, k-nearest neighbours (KNN), multi-layer perceptron (MLP), random forest, and XGBoost. Credit card loans are a type of unsecured debt whereby the loan is typically granted solely based on the customer's credit history and personal characteristics [Kim, Cho, and Ryu, 2018].

Therefore, using a well-regarded cross-sectional credit card default dataset, we will investigate the ability of the models to predict whether customers will default on their credit card payment within the next month.

1.1 Literature Review

As Hand and Henley [1997] observes, the earliest credit-scoring methods were largely based on human judgement informed by past outcomes. However, the rapid growth of consumer credit and rising interbank competition spurred the adoption of statistical models for assessing individuals' creditworthiness. Early examples included linear discriminant analysis (LDA), probit models, logistic regression, and linear probability models (see Hand and Henley [1997], Thomas [2000] for detailed surveys). A common criticism of these models is that many of their underlying assumptions are often violated, undermining their theoretical validity [Wang et al., 2011]. Moreover, because they are linear, these models struggle to capture complex, non-linear relationships among features.

With rising computational power and the proliferation of more sophisticated machine learning methods in recent decades, researchers have observed that traditional approaches are often outperformed by more complex models [Bhandary and Ghosh, 2025]. Yeh and Lien [2009] employ a Taiwanese consumer dataset to compare six key machine learning techniques in terms of classification performance and predictive accuracy for the probability of default. They find that, despite small differences in error rates, artificial neural networks (ANNs) deliver the most accurate classifications. However, it is worth noting that their study did not use decision tree ensemble algorithms such as XGBoost and random forests, which have proven highly successful in this classification task.

Wang et al. [2011] examine the effectiveness of three ensemble methods – bagging, boosting, and stacking – for classifying customers as 'good' or 'bad' credit. They report that all three ensemble methods substantially outperform individual learners. Likewise, Alam et al. [2020] use three prominent credit default datasets to show that gradient-boosted decision tree methods achieve better results than traditional machine learning approaches. Moreover, Xia et al. [2020] demonstrate that CatBoost, another advanced gradient-boosting decision tree algorithm, outperforms the industry standard for predicting loan defaults in the peer-to-peer lending market. A further large-scale benchmarking study by Lessmann et al. [2015] – spanning eight datasets and 41 different classifiers – finds that heterogeneous ensemble classifiers perform very well, surpassing even homogeneous ensemble methods.

Many prominent datasets that are publicly available in the credit scoring domain are cross-sectional in nature due to data collection complexities and privacy concerns. However, research that is conducted on both cross-sectional and panel datasets highlights the success of machine learning models over the traditional methods. Butaru et al. [2016] analyse a panel dataset consisting of aggregated data from six large US banks and illustrate the effectiveness of machine learning techniques. Their results conclude that decision trees and random forests outperform more traditional methods, such as logistic regression, in both in-sample and out-of-sample prediction of credit card delinquencies. Researchers argue that to fully understand the default risk of an individual, dynamic models need to be employed [Thomas, Ho, and Scherer, 2001]. One key reason for this is that dynamic models enable changes in characteristics and the macroeconomic environments to be taken into consideration [Kim, Cho, and Ryu, 2018].

2 Methodology

2.1 Baseline Models

2.1.1 Random Classifier

Random classifiers use a stratified strategy that generates predictions with respect to the distribution of the response variable, while not taking into account any of the features. The models we pick should be meaningful enough to outperform a random guess, especially one that does not take features into account. Another aim of this model is to highlight issues surrounding imbalance within the data. In the case of imbalanced data, random classifiers may show strong accuracy by simply guessing the majority class most of the time. Evaluating their scores using the F_1 score, AUC, precision, and recall metrics and comparing with our chosen models shows their effectiveness in identifying rare

events. Given that our response variable is split 2:7 defaulted loans to non-defaulted loans, a random classifier is an ideal baseline to implement.

2.1.2 Naive Classifier

Another baseline model we will implement is a mode-based naive classifier that simply assigns each instance to the most frequent class, in this case, non-defaulted loans. The extremely simple nature of this classifier is intended to establish whether our models can truly learn from the selected features. In scenarios where the model barely outperforms this naive approach, it may be inferred that any performance gain is merely due to the class imbalances rather than the predictive power of our features.

This combination of baselines aims to ensure that our feature selection is robust enough to yield justifiable predictions that are not unduly influenced by imbalances in the data. They further guarantee that any improvements in model performance extend beyond what could be achieved by random prediction strategies.

2.2 Classification Models

2.2.1 K-Nearest Neighbours (KNN)

The KNN classification method assigns class labels by selecting the k-nearest samples and then assigning the input to the majority class among those samples using a distance metric, usually the Euclidean distance:

$$d(X, Y) = \sqrt{\sum_{i=1}^{N} (X_i - Y_i)^2} \tag{1}$$

Given that KNN does not assume any underlying data distribution, it is well suited to handling non-linear decision boundaries. Credit default data is inherently imbalanced, making the choice of k (the number of nearest neighbours considered) a sensitive hyperparameter. An inappropriate selection of k can lead to issues such as a high number of false negatives. Specifically, a value of k that is too high may reinforce the class imbalance, while a value that is too low can result in noisy predictions. By employing synthetic minority oversampling techniques (SMOTE), we can better adjust the data to reduce this imbalance. Consequently, we have set $k = 50$ to achieve better generalisation.

2.2.2 Random Forest

The random forest classification algorithm is an ensemble algorithm that builds a group of decision trees sequentially during the training phase, and outputs the mode of each class. Each tree within the ensemble is trained on a bootstrap sample of the training data allowing for diversity in data that the model is trained on in each iteration, promoting greater generalisation for unseen data. Random forests hold large strength in classification tasks primarily due to their ability to find complex non-linear relationships in large sets of tabular, structured data that may be imbalanced. The final prediction is then obtained through a majority vote across all trees where $f^{(t)}(x)$ is the majority voted output for the t^{th} tree:

$$\hat{y} = \text{mode}\{f_1(x), f_2(x), \ldots, f_T(x)\}$$

2.2.3 Logistic Regression

Logistic regression is a binary classification model that estimates a probability that an input belongs to the positive case (1). This is achieved using a logistic function that maps the input into a range between 0 and 1. The probability that an input belongs to a positive case is given by:

$$P(y = 1 \mid X) = \left(1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k)}\right)^{-1} \tag{2}$$

Where β_i are optimised by minimising the log-loss function:

$$L\left(y, \hat{y}\right) = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log(\hat{y}_i) + (1 - y_i) \log\big(1 - \hat{y}_i\big) \Big] \tag{3}$$

A threshold is then set between zero and one, which is then used to assign a class label with the probability formula at the top:

$$\hat{y} = \begin{cases} 1 & \text{if } P\big(y = 1 \mid X\big) \geq 0.5, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Logistic regression can therefore provide clear, interpretable relationships between the features and probability of default. However, due to its linear decision boundary it struggles with finding non-linear patterns in data. This is dissimilar to models that use non-linear decision boundaries such as KNN. KNN has a decision boundary shaped by how the data clusters locally.

2.2.4 XGBoost

XGBoost is a gradient boosting algorithm which takes an ensemble of regression and classification trees grouped sequentially and iterates through them. In every iteration across the trees, the objective function, which we aim to minimise, contains a loss function and a regularisation term that prevents overfitting, and allows for greater generalisation to unseen data.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k) \tag{5}$$

In every iteration, the model computes the residual errors which is the difference between the true values and current predicted value, and a new tree is then trained to predict these residuals in an attempt to 'correct' the errors by the previous ensemble. This new tree is then added to the model and new predictions are outputted, and this continues to iterate until either convergence occurs, or a fixed number of iterations are completed.

Within the objective function, the loss function measures how well the model fits the training data. The regularisation term works to penalise tree complexity as to push preference towards simpler trees for better generalisation.

XGBoost is capable of capturing complex, non-linear relationships among features, similar to other non-linear models such as KNN. Moreover, its ability to compute feature importance enhances interpretability by identifying which features significantly influence the probability of default. Additionally, XGBoost exhibits resistance to overfitting, which is crucial for robust generalisation, and it often outperforms random forests and other models that are more prone to overfitting.

2.2.5 Multi-Layer Perceptron (MLP)

An MLP is a type of feed-forward neural network which is fully connected and has at least one hidden layer. MLPs are able to learn complex non-linear relationships that are mapped between input features and outputs, making them a strong choice for our problem.

The forward pass in the MLP takes the form:

$$Z = Wx + b \tag{6}$$

In this linear combination; W represents the weights, X are the inputs, b is the bias, and σ is the activation function. This linear equation is then passed through an activation function, in our case the sigmoid function given that we are implementing a binary classification:

$$a = \sigma(z) \tag{7}$$

The sigmoid function then maps the output to an interval between zero and one, which then identifies the probability of being assigned a positive class (1).

Due to the nature of the project being a binary classification, we utilise a binary cross-entropy loss function to be minimised:

$$L\left(y, \hat{y}\right) = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log(\hat{y}_i) + (1 - y_i) \log\big(1 - \hat{y}_i\big) \Big] \tag{8}$$

In our implementation, we used a network architecture with two hidden layers. The first hidden layer comprises 100 neurons, while the second has 50 neurons. The model was trained over 1000 epochs with a batch-based algorithm as a way to improve prediction accuracy.

2.3 Dataset

The data was sourced from Yeh and Lien [2009], which is a well-regarded data set for credit modelling. It tracks whether a card holder defaults on their payment (Yes = 1; No = 0) using data that covers past payments – along with other features – from April 2005 to September 2005. Using this data, our classifier looks to predict whether a card holder defaults in October 2005. The target variable is binary where a value of 1 indicates a credit card default, while 0 indicates no default.

The dataset encompasses both financial and non-financial features. The financial features include *Amount of Credit Given*, *Amount of Bill Statement*, and *Amount of Previous Payment*, all denominated in New Taiwan (NT) dollars. It also tracks the *History of Past Payments*, where -1 indicates a timely payment, and any positive integer n represents a payment delay of n months.

Alongside these financial features, the dataset contains non-financial data regarding card holders: *Gender* (1 = male; 2 = female), *Education* (1 = graduate school; 2 = university; 3 = high school; 4 = other), *Marital Status* (1 = married; 2 = single; 3 = other), and *Age*. Each of these may influence the likelihood of default.

2.3.1 Data Cleaning

We conducted an initial exploration of the data. First, we checked for null values in any of the features, to which there were none. Next, we removed entries in *Education and Marriage* that are not classified by our system, then recoded all undocumented values in the repayment columns (PAY_x) as zero. We found that the *Amount of Bill Statement* can be negative; this indicates credit in the card holder's favour, possibly due to overpayment on their last bill. Finally, since the data already has the appropriate labels for our experiment, we made sure that the default column was left out of the feature set to prevent leakage. At this stage, the data was deemed usable.

The one-hot encoder converts categorical variables with multiple categories into several binary columns. This method is preferable to label encoding, which implicitly imposes an ordinal relationship. A robust scaler is used because it is more appropriate for financial data, where distributions are often non-normal. This scaler removes the median and scales the data based on the quantile range (defaulting to the interquartile range, or IQR). The IQR is the span between the 1st quartile (25th percentile) and the 3rd quartile (75th percentile). Finally, two versions of the dataset were created: a scaled one for models requiring scaled data, and an unscaled one for models that do not.

2.4 Synthetic Minority Over-sampling Technique (SMOTE)

Upon exploring the data distribution, we observed that the dataset is highly imbalanced. Only 22.313% of cardholders defaulted on their payment in October 2005, leaving 77.686% who did not. This imbalance poses a challenge for classification models, as they may become biased towards predicting the majority class (non-default). Although a model might appear to perform well in terms of accuracy, it would likely fail to correctly identify the minority class (defaults), which is of greater concern in this context.

To address this, we applied SMOTE [Chawla et al., 2002] during preprocessing. SMOTE generates synthetic examples of the minority class by interpolating between existing examples. For each sample x_i in the minority class SMOTE selects one of its k-nearest neighbours x_{nn}, then creates a synthetic point x_{new} by linear interpolation:

$$x_{\text{new}} = x_i + \lambda(x_{\text{nn}} - x_i) \tag{9}$$

Where x_i is the feature vector of a minority class sample, x_{nn} is one of its k-nearest neighbours and $\lambda \sim U(0, 1)$. This process is repeated to generate enough synthetic samples to equalise the number of samples in both classes, thereby balancing the dataset. This helps to prevent the model from being biased towards the majority class.

In our analysis, we trained and evaluated each model on both the original (imbalanced) dataset and the SMOTE-balanced dataset. This enabled a direct comparison of performance metrics and provided insight into how class balancing affects the classifier's ability to correctly identify defaulters.

2.5 Model Accuracy

To evaluate the performance of our predictions, we used a combination of four basic metrics on out-of-sample data.

2.5.1 Balanced Accuracy

Balanced accuracy is the average recall value for each class. It adjusts the standard accuracy score to account for class imbalance, by giving equal weight to each class, so it better reflects true performance:

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\right) \tag{10}$$

Where, TP are the true positives, TN are the true negatives, FP are the false positives and FN are the false negatives.

2.5.2 Weighted F_1 Score

The weighted F_1 score is the weighted average of the F_1 scores for each class. For binary classification, the F_1 score is the harmonic mean of the precision and recall:

$$\text{F}_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \quad \text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{11}$$

This metric accounts for both how many predicted defaulters are actually defaulters (precision) and how many real defaulters are correctly identified (recall). In loan default prediction, it is particularly important because misclassifying an actual defaulter as low-risk can be far more costly than the reverse.

The weighted F_1 score further adjusts for class imbalance by averaging F_1 scores for each class according to their frequency. This makes it especially useful in scenarios like ours, where the number of defaults is much smaller than non-defaults, but correctly identifying them is critical for financial risk management.

2.5.3 Receiver Operating Characteristic (ROC) Curve

The ROC curve is a plot of the *true positive rate (TPR)* against the *false positive rate (FPR)* across different classification thresholds:

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}} \tag{12}$$

An ROC curve that lies above the diagonal line represents a model that is better than random. A perfect model hugs the top-left corner. The *area under the curve (AUC-ROC)* is a numerical summary of the ROC curve, where 1 indicates perfect performance, 0.5 indicates random guessing, and less than 0.5 indicates worse-than-random predictions. AUC-ROC is unique to binary classification and is excellent at measuring its success. It is particularly useful for imbalanced data, as it shows how well the model separates the classes.

2.5.4 Matthews Correlation Coefficient (MCC)

The MCC is a robust metric for binary classification that considers all four elements of the confusion matrix:

$$\text{MCC} = \frac{(\text{TP} \cdot \text{TN}) - (\text{FP} \cdot \text{FN})}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{13}$$

MCC ranges from +1 (perfect prediction), through 0 (no better than random), to -1 (perfect disagreement). We decided to use MCC because it punishes all types of errors equally, making it especially valuable in imbalanced classification settings. MCC will be useful for measuring how the rebalancing of data affects the predictions of the models.

3 Results

We assessed out-of-sample performance on the full dataset, to simulate default prediction in a production environment. Then we applied k-fold cross-validation to evaluate each model's ability to generalise to unseen data. We also evaluated the model across both the imbalanced data and SMOTE-balanced data.

3.1 Out-of-Sample Performance

We used an 80/20 train-test split. Tables 1a & 1b summarise the performance across all models.

Table 1: Performance summary across models and data.

(a) Original (imbalanced) data

Model	Accuracy	Weighted F_1	AUC-ROC	MCC
Random	0.4908	0.6466	0.4908	-0.0186
Naive	0.5000	0.6773	0.5000	0.0000
Logistic Regression	0.6304	0.7802	0.7550	0.3535
KNN	0.6308	0.7795	0.7617	0.3498
MLP	0.6355	0.7667	0.7061	0.3063
Random Forest	0.6388	0.7826	0.7690	0.3564
XGB	0.6365	0.7800	0.7532	0.3469

(b) SMOTE-Balanced data

Model	Accuracy	Weighted F_1	AUC-ROC	MCC
Random	0.5088	0.5459	0.5088	0.0146
Naive	0.5000	0.6773	0.5000	0.0000
Logistic Regression	0.6988	0.7805	0.7533	0.3834
KNN	0.6881	0.7152	0.7543	0.3220
MLP	0.6306	0.7251	0.6849	0.2445
Random Forest	0.6580	0.7807	0.7527	0.3505
XGB	0.6558	0.7832	0.7472	0.3561

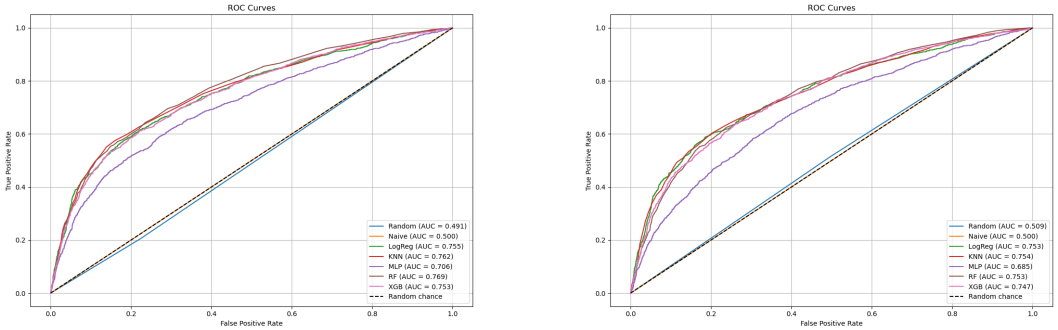
In both datasets, the advanced classifiers significantly outperformed the random and naive baselines, with logistic regression, XGBoost, and random forest demonstrating the strongest overall performance. The random and naive classifiers exhibited poor balanced accuracy and AUC-ROC ($\sim$0.50), indicating near-random predictions, and yielded MCC values close to zero, suggesting no effective learning. Although their weighted F_1 scores might appear indicative of reasonable performance, they primarily reflect the heavy weighting of the majority class. Consequently, these models fail to accurately identify defaulters and are biased towards the majority class. This shortcoming is evident in the marked ($\sim$ %10) drop in weighted F_1 when moving from imbalanced to balanced data. The naive model's F_1 score remained unchanged because it always predicts the majority class, and SMOTE was only applied to the training set, leaving the test set unaffected.

For the imbalanced data, logistic regression, KNN, XGBoost, and random forest achieved the highest overall performance, reporting similar results: balanced accuracies of approximately 0.63, weighted F_1 scores of around 0.77–0.78, AUC-ROC values of about 0.75–0.76, and MCCs of 0.34–0.35. These metrics point to strong discriminative ability and, through MCC, indicate that the models effectively learnt the underlying data patterns. By contrast, the MLP exhibited comparable balanced accuracy, a slightly lower weighted F_1 ($\sim$0.77), but noticeably lower AUC-ROC (0.70) and MCC (0.30). This reduction implies a higher incidence of false positives and fewer true positives relative to the other models.

When trained on the SMOTE-balanced data, overall model performance changed only slightly. Logistic regression remained robust, improving on most metrics: its balanced accuracy increased to around 0.70, and its MCC rose to 0.38. This suggests that logistic regression was already relatively resilient to class imbalance and derived modest benefit from balancing. XGBoost and random forest also maintained strong F_1 and AUC-ROC scores (both above 0.75) and MCC values exceeding 0.35, indicating reliable out-of-sample generalisation.

Notably, KNN showed a decrease in its F_1 score (from approximately 0.78 to 0.71) after balancing, while its AUC-ROC and MCC remained relatively stable. This finding implies an increased rate of false positives despite enhanced discriminative capability. Figures 1a and 1b reveal a marked ($\sim$0.2) decline in MLP performance between the two datasets. MLP also exhibited lower F_1 and MCC after balancing, possibly reflecting heightened sensitivity to the altered class priors introduced by SMOTE.

Overall, these results confirm that logistic regression, random forest, and XGBoost consistently deliver strong and balanced out-of-sample performance, with logistic regression emerging as the best-performing model. Their stability across different class distributions and performance metrics underlines their suitability for credit default prediction tasks in imbalanced domains.



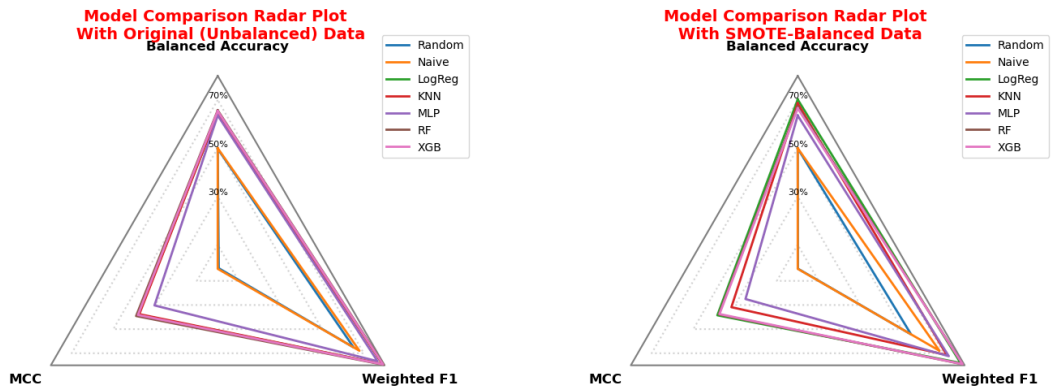
(a) Original (imbalanced) data

(b) SMOTE-balanced data

Figure 1: Comparative ROC curves across models. Greater AUC indicates better performance.

3.2 k-Fold Cross Validation

For k-fold cross-validation, the dataset is divided into k folds (or subsets). Each fold is a random partition, where each datapoint appears in exactly one fold. For each of the k rounds, a model is trained on $k-1$ folds and tested on the remaining fold. This process is repeated k times, with each fold used once as the validation set. The final k-fold cross-validation result is the mean performance across all k rounds. Given the size of the dataset, we opted for 10 folds. (Tables 2 & 3 in Appendix A show model performance across validations)



(a) Original (imbalanced) data

(b) SMOTE-balanced data

Figure 2: Radar plots of the mean balanced accuracy, weighted F_1, and MCC across 10 validations. Higher scores (larger areas) indicate better performance.

Logistic regression consistently demonstrated strong, stable performance across folds, with the highest MCC and balanced accuracy on SMOTE-balanced data, indicating good predictive power across both datasets. The ensemble methods, random forest and XGBoost, performed similarly well across both datasets, though their improvements with SMOTE were less pronounced, leaving them behind the logistic regression model. This is likely due to the ensemble models handling the imbalanced data more effectively, owing to their robustness and flexibility.

KNN lost performance after balancing, with a decrease of approximately 0.60 and 0.07 in the MCC and F_1 scores respectively. This mirrors the differences seen in out-of-sample performance but with a more noticeable drop in MCC. The MLP also declined in performance on balanced data, with a reduction of approximately 0.05 in MCC and 0.04 in the F_1 score. This may indicate that these models are more sensitive to changes in data distribution introduced by synthetic sampling, potentially due to their reliance on local structure (in the case of KNN) or over-fitting to training patterns (in the case of MLP).

Our baseline models performed as expected and confirmed that the trained models exhibit noticeable performance improvements. Overall, these results validated the use of logistic regression as the best model on resampled data, and random forest or XGBoost on imbalanced data. All three of these models where shown to be robust, generalisable choices for this classification task – particularly when addressing class imbalance.

4 Discussion

This study evaluated a variety of classification algorithms for credit default prediction across both imbalanced and SMOTE-balanced datasets, using out-of-sample testing and k-fold cross-validation to ensure robust performance assessment. Across all evaluation methods, advanced classifiers significantly outperformed the random and naive baselines, confirming the benefit of model-based learning in highly imbalanced domains.

Logistic regression, XGBoost, and random forest consistently emerged as the models with the highest performance, demonstrating strong generalisation, high balanced accuracy, weighted F_1, AUC-ROC, and MCC. Notably, logistic regression showed the most stable performance across both class distributions and evaluation methods, with particular gains under SMOTE-balancing, highlighting its robustness to class imbalance and suitability for deployment in real-world systems where reliability is critical.

XGBoost and random forest also performed well, particularly on imbalanced data, supporting their reputation for robustness and flexibility in noisy or skewed settings. However, these models benefited less from synthetic balancing, suggesting diminishing returns from oversampling when using ensemble methods.

In contrast, KNN and MLP showed notable sensitivity to the data distribution. KNN experienced significant declines in F_1 and MCC after SMOTE-balancing, likely due to increased false positives from altered local structures. Similarly, MLP performance deteriorated, possibly due to overfitting to synthetic training patterns. These results highlight the importance of model stability and sensitivity to class-balancing techniques. The relative decline in performance for models like KNN and MLP suggests that some classifiers are more affected by the distributional shifts introduced by SMOTE. This reinforces the need to evaluate models not just on headline metrics, but also on their robustness to preprocessing strategies when dealing with imbalanced data.

Overall, the results support the use of logistic regression as a reliable and interpretable baseline for default prediction, especially on resampled data. Ensemble methods such as random forest and XGBoost offer strong alternatives, particularly in scenarios where resampling is not feasible. These three models provide robust and generalisable solutions for credit default classification and are recommended for deployment in imbalanced financial datasets.

With respect to future implementation, this study is limited by its use of a relatively small dataset consisting of Taiwanese credit card clients from a single financial institution. The data, collected in 2005, may not reflect current financial behaviours or macroeconomic conditions, and lacks a time-series component that would more accurately mirror real-world credit risk assessment. The binary classification based on a fixed six-month window leading to a single default decision in the seventh month oversimplifies the dynamic and continuous nature of credit risk.

Importantly, this study focused on how resampling affects different models, but was limited to a single oversampling method – SMOTE. Future work could investigate more advanced or targeted techniques such as ADASYN (Adaptive Synthetic Sampling), or SMOTE variations like Borderline-SMOTE and Cluster-SMOTE, which generate synthetic samples with greater sensitivity to decision boundaries. Hybrid approaches, such as combining SMOTE with Tomek Links, offer a promising direction by coupling oversampling with the removal of noisy or ambiguous instances through undersampling. Exploring these alternatives could yield further insights into how different resampling strategies influence classifier performance. Overall, this study underscores the significant impact resampling can have on model evaluation and highlights the importance of a nuanced understanding of preprocessing techniques in imbalanced learning scenarios.

References

Federal Reserve Bank of New York. Household debt balances continue steady increase; delinquency transition rates remain elevated for auto and credit cards, feb 2025. URL https://www.newyorkfed.org/newsevents/news/research/2025/20250213. Press release. Accessed: 03 April 2025.

Federal Reserve Bank of St. Louis. Delinquency rate on credit card loans, all commercial banks, 2025. URL https://fred.stlouisfed.org/series/DRCCLACBS. Accessed: April 2, 2025.

Financial Times. Us credit card defaults jump to highest level since 2010, 2025. URL https://www.ft.com/content/c755a34d-eb97-40d1-b780-ae2e2f0e7ad9. Accessed: 03 April 2025.

Talha Mahboob Alam, Kamran Shaukat, Ibrahim A Hameed, Suhuai Luo, Muhammad Umer Sarwar, Shakir Shabbir, Jiaming Li, and Matloob Khushi. An investigation of credit card default prediction in the imbalanced datasets. *Ieee Access*, 8:201173–201198, 2020.

Hyeongjun Kim, Hoon Cho, and Doojin Ryu. An empirical study on credit card loan delinquency. *Economic systems*, 42(3):437–449, 2018.

D. J. Hand and W. E. Henley. Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 160(3):523–541, 1997. ISSN 09641998, 1467985X. URL http://www.jstor.org/stable/2983268.

Lyn C. Thomas. A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting*, 16(2):149–172, 2000. ISSN 0169-2070. doi: https://doi.org/10.1016/S0169-2070(00)00034-0. URL https://www.sciencedirect.com/science/article/pii/S0169207000000340.

Gang Wang, Jinxing Hao, Jian Ma, and Hongbing Jiang. A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 38(1):223–230, 2011. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2010.06.048. URL https://www.sciencedirect.com/science/article/pii/S095741741000552X.

Rakshith Bhandary and Bidyut Kumar Ghosh. Credit card default prediction: An empirical analysis on predictive performance using statistical and machine learning methods. *Journal of Risk and Financial Management*, 18(1):23, 2025.

I-Cheng Yeh and Che-Hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2, Part 1):2473–2480, 2009. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2007.12.020. URL https://www.sciencedirect.com/science/article/pii/S0957417407006719.

Yufei Xia, Lingyun He, Yinguo Li, Nana Liu, and Yanlin Ding. Predicting loan default in peer-to-peer lending using narrative data. *Journal of Forecasting*, 39(2):260–280, 2020.

Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, and Lyn C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2015.05.030. URL https://www.sciencedirect.com/science/article/pii/S0377221715004208.

Florentin Butaru, Qingqing Chen, Brian Clark, Sanmay Das, Andrew W. Lo, and Akhtar Siddique. Risk and risk management in the credit card industry. *Journal of Banking & Finance*, 72:218–239, 2016. ISSN 0378-4266. doi: https://doi.org/10.1016/j.jbankfin.2016.07.015. URL https://www.sciencedirect.com/science/article/pii/S0378426616301340.

Lyn C Thomas, Joseph Ho, and William T Scherer. Time will tell: behavioural scoring and the dynamics of consumer credit assessment. *IMA Journal of Management Mathematics*, 12(1):89–103, 2001.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. doi: 10.1613/jair.953. URL https://doi.org/10.1613/jair.953.

A Appendices

A.1 A: K-fold Cross Validation Performance

Table 2: Cross-validation results (10 folds) across models for original (imbalanced) data.

Model	Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean ± Std
Random	AUC-ROC	0.5132	0.5136	0.4849	0.5030	0.5052	0.5008	0.4929	0.4935	0.4861	0.4818	0.4975 ± 0.0109
	Balanced Accuracy	0.5132	0.5136	0.4849	0.5030	0.5052	0.5008	0.4929	0.4935	0.4861	0.4818	0.4975 ± 0.0109
	MCC	0.0265	0.0273	-0.0306	0.0062	0.0104	0.0015	-0.0139	-0.0128	-0.0276	-0.0359	-0.0049 ± 0.0217
	Weighted F_1	0.6607	0.6615	0.6334	0.6408	0.6548	0.6630	0.6527	0.6551	0.6449	0.6449	0.6512 ± 0.0093
Naive	AUC-ROC	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000 ± 0.0000
	Balanced Accuracy	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000 ± 0.0000
	MCC	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000 ± 0.0000
	Weighted F_1	0.6769	0.6777	0.6648	0.6552	0.6763	0.6957	0.6874	0.6906	0.6818	0.6869	0.6793 ± 0.0116
Logistic Regression	AUC-ROC	0.7519	0.7608	0.7665	0.7568	0.7471	0.7736	0.7479	0.7534	0.7598	0.7548	0.7573 ± 0.0078
	Balanced Accuracy	0.6165	0.6476	0.6529	0.6389	0.6450	0.6588	0.6437	0.6317	0.6453	0.6517	0.6432 ± 0.0114
	MCC	0.3220	0.3893	0.3886	0.3730	0.3903	0.4077	0.3870	0.3585	0.3884	0.3876	0.3792 ± 0.0225
	Weighted F_1	0.7701	0.7920	0.7860	0.7737	0.7908	0.8082	0.7961	0.7894	0.7936	0.7975	0.7897 ± 0.0106
KNN	AUC-ROC	0.7484	0.7765	0.7659	0.7481	0.7581	0.7860	0.7632	0.7363	0.7488	0.7576	0.7589 ± 0.0140
	Balanced Accuracy	0.6236	0.6496	0.6498	0.6426	0.6455	0.6543	0.6380	0.6357	0.6527	0.6477	0.6439 ± 0.0089
	MCC	0.3357	0.3898	0.3748	0.3712	0.3770	0.3995	0.3721	0.3551	0.4022	0.3816	0.3759 ± 0.0189
	Weighted F_1	0.7746	0.7925	0.7821	0.7744	0.7881	0.8056	0.7917	0.7895	0.7982	0.7955	0.7892 ± 0.0094
MLP	AUC-ROC	0.7018	0.7121	0.7101	0.7019	0.6972	0.7053	0.7054	0.7190	0.7062	0.6933	0.7052 ± 0.0070
	Balanced Accuracy	0.6433	0.6270	0.6409	0.6294	0.6417	0.6303	0.6336	0.6487	0.6345	0.6297	0.6359 ± 0.0069
	MCC	0.2899	0.3036	0.3143	0.2893	0.3057	0.3052	0.3047	0.3178	0.2969	0.3039	0.3031 ± 0.0087
	Weighted F_1	0.7537	0.7671	0.7621	0.7483	0.7641	0.7779	0.7723	0.7756	0.7654	0.7724	0.7659 ± 0.0089
Random Forest	AUC-ROC	0.7523	0.7831	0.7650	0.7509	0.7666	0.7980	0.7582	0.7568	0.7522	0.7690	0.7652 ± 0.0144
	Balanced Accuracy	0.6395	0.6490	0.6631	0.6468	0.6598	0.6694	0.6588	0.6572	0.6692	0.6587	0.6572 ± 0.0091
	MCC	0.3605	0.3807	0.3912	0.3661	0.3988	0.4162	0.3964	0.3937	0.4171	0.3905	0.3911 ± 0.0175
	Weighted F_1	0.7835	0.7904	0.7882	0.7742	0.7958	0.8116	0.8010	0.8018	0.8046	0.7992	0.7950 ± 0.0105
XGB	AUC-ROC	0.7499	0.7748	0.7711	0.7506	0.7712	0.7792	0.7527	0.7496	0.7531	0.7645	0.7617 ± 0.0111
	Balanced Accuracy	0.6362	0.6462	0.6714	0.6472	0.6599	0.6676	0.6547	0.6471	0.6563	0.6530	0.6540 ± 0.0100
	MCC	0.3471	0.3759	0.4055	0.3725	0.3933	0.4122	0.3841	0.3663	0.3871	0.3791	0.3823 ± 0.0179
	Weighted F_1	0.7797	0.7887	0.7930	0.7758	0.7944	0.8104	0.7973	0.7937	0.7953	0.7956	0.7924 ± 0.0091

Table 3: Cross-validation results (10 folds) across models for SMOTE-balanced data.

Model	Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean ± Std
Random	AUC-ROC	0.5188	0.4926	0.4926	0.4877	0.4916	0.5028	0.5028	0.5116	0.5077	0.4853	0.4994 ± 0.0105
	Balanced Accuracy	0.5188	0.4926	0.4926	0.4877	0.4916	0.5028	0.5028	0.5116	0.5077	0.4853	0.4994 ± 0.0105
	MCC	0.0313	-0.0123	-0.0123	-0.0204	-0.0139	0.0047	0.0047	0.0193	0.0128	-0.0245	-0.0011 ± 0.0175
	Weighted F_1	0.5557	0.5391	0.5391	0.5360	0.5385	0.5457	0.5457	0.5513	0.5488	0.5345	0.5434 ± 0.0067
Naive	AUC-ROC	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000 ± 0.0000
	Balanced Accuracy	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000 ± 0.0000
	MCC	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000 ± 0.0000
	Weighted F_1	0.6792	0.6791	0.6791	0.6791	0.6791	0.6795	0.6795	0.6795	0.6795	0.6795	0.6793 ± 0.0002
Logistic Regression	AUC-ROC	0.7763	0.7570	0.7635	0.7657	0.7381	0.7449	0.7588	0.7722	0.7397	0.7589	0.7575 ± 0.0124
	Balanced Accuracy	0.7162	0.7084	0.6984	0.6989	0.6951	0.6899	0.7156	0.7160	0.6879	0.6919	0.7018 ± 0.0107
	MCC	0.4032	0.3978	0.3786	0.3829	0.3797	0.3649	0.4104	0.4070	0.3583	0.3687	0.3852 ± 0.0176
	Weighted F_1	0.7823	0.7848	0.7780	0.7809	0.7814	0.7745	0.7891	0.7859	0.7708	0.7758	0.7804 ± 0.0053
KNN	AUC-ROC	0.7631	0.7578	0.7592	0.7575	0.7466	0.7504	0.7523	0.7656	0.7443	0.7475	0.7544 ± 0.0069
	Balanced Accuracy	0.7013	0.6843	0.6849	0.6909	0.6831	0.6781	0.6793	0.6995	0.6765	0.6816	0.6860 ± 0.0082
	MCC	0.3413	0.3135	0.3199	0.3281	0.3164	0.3054	0.3052	0.3392	0.3003	0.3098	0.3179 ± 0.0135
	Weighted F_1	0.7159	0.7083	0.7239	0.7225	0.7216	0.7126	0.7066	0.7180	0.7042	0.7104	0.7144 ± 0.0067
MLP	AUC-ROC	0.7025	0.6822	0.6924	0.6877	0.6960	0.6931	0.6939	0.6910	0.6551	0.6791	0.6873 ± 0.0125
	Balanced Accuracy	0.6496	0.6347	0.6395	0.6313	0.6410	0.6330	0.6460	0.6356	0.6090	0.6350	0.6355 ± 0.0104
	MCC	0.2626	0.2484	0.2593	0.2502	0.2525	0.2549	0.2795	0.2483	0.2038	0.2446	0.2504 ± 0.0182
	Weighted F_1	0.7107	0.7237	0.7298	0.7320	0.7157	0.7351	0.7436	0.7219	0.7114	0.7175	0.7241 ± 0.0103
Random Forest	AUC-ROC	0.7690	0.7640	0.7621	0.7563	0.7481	0.7564	0.7509	0.7603	0.7421	0.7465	0.7556 ± 0.0081
	Balanced Accuracy	0.6833	0.6677	0.6836	0.6614	0.6648	0.6724	0.6846	0.6723	0.6612	0.6675	0.6719 ± 0.0086
	MCC	0.3951	0.3808	0.4057	0.3578	0.3703	0.3822	0.4003	0.3752	0.3496	0.3662	0.3783 ± 0.0173
	Weighted F_1	0.7954	0.7921	0.7997	0.7841	0.7885	0.7923	0.7976	0.7894	0.7807	0.7866	0.7906 ± 0.0057
XGB	AUC-ROC	0.7678	0.7492	0.7610	0.7556	0.7508	0.7482	0.7548	0.7660	0.7511	0.7561	0.7560 ± 0.0065
	Balanced Accuracy	0.6720	0.6663	0.6797	0.6700	0.6523	0.6568	0.6676	0.6574	0.6610	0.6697	0.6653 ± 0.0079
	MCC	0.3799	0.3770	0.4102	0.3783	0.3577	0.3582	0.3857	0.3617	0.3587	0.3820	0.3749 ± 0.0157
	Weighted F_1	0.7913	0.7908	0.8017	0.7909	0.7850	0.7851	0.7941	0.7863	0.7847	0.7926	0.7902 ± 0.0050