Can defaulting of a client be predicted using client details and past transactions?

Akshat Gupta, AkshatG0104@gmail.com

1 Introduction

In the complex landscape of financial services, credit card defaults are a crucial challenge with profound implications for consumers and financial institutions. This report presents a comprehensive summary of a machine learning model that attempts to predict whether a credit card client will default based on various features such as historical transactions.

Credit card defaulting is when clients fail to meet the minimum payment required on their credit card accounts. For financial institutions, a client defaulting leads to financial losses and many other negative implications. For consumers, defaulting results in a substantial decrease in credit scores, financial strains, and many other negative implications.

The applications of a model that predicts whether a credit card client will default obviously promise several real-world benefits. The application of such a model allows financial institutions to identify high-risk individuals, enabling more informed measures and enforcing effective risk management decisions. Furthermore, financial institutions can use such a model to aid vulnerable consumers and offer appropriate financial advice.

1.1 Problem definition

This machine learning problem is framed as a supervised binary classification problem. The efficacy of the model will be evaluated based on its accuracy, precision, recall and F1-score. The input for the model are 23 feature variables and the output is a target 1(Client defaults) or 0(Client does not default).
The following figure is what I propose as the Machine Learning system that should be used to preprocess the data and train the model for the best results.

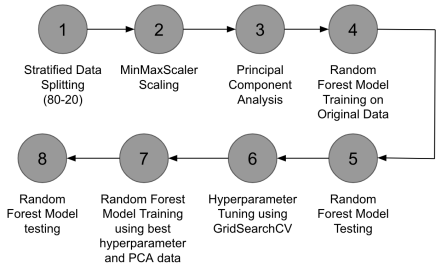

Fig. 1. Proposed Machine Learning System

This report gives an in-depth analysis of the reasons behind the recommended Machine Learning system.

2 Dataset

The model training utilizes the 'Default of Credit Card Clients' dataset, comprising 23 feature variables and 1 target variable. Details about each variable, including inputs and data cleaning methods, are outlined in the TABLE 1.

TABLE 1
Dataset Variables Description

Variable	Variable Description	Data Cleaning
ID	ID of each client	N/A
LIMIT_BAL	Amount of given credit (NT dollars)	N/A
SEX	1 = Male, 2 = Female	N/A
EDUCATION	1 = Graduate School, 2 = University, 3 = High School, 4 = Others	Removed all instances that contained invalid inputs (0, 5, 6)
MARRIAGE	1 = Married, 2 = Single, 3 = Others	Removed all instances that contained invalid inputs (0)
AGE	Age in years	N/A
PAY_0 to PAY_6	History of past payment for 6 months. -1 = Pay duly, 1-9 = Payment delay(1 = 1 month, 9 = 9 month)	Changed inputs such as -2 and 0 to -1
BILL_AMT1 to BILL_AMT6	Amount of bill statement (NT dollar)	N/A
PAY_AMT1 to PAY_AMT6	Amount of previous payment (NT dollar)	N/A
DEFAULT(Target Variable)	Client defaults next month. 1 = Yes, 0 = No	N/A

For the 'MARRIAGE' and 'EDUCATION' columns, instances with invalid inputs were removed from the dataset. This decision was predicated on the fact that such instances constituted a mere 1.33%. This exclusion of these instances enhanced the overall clarity of the dataset without impacting the integrity of the dataset significantly.

2.1 Data Exploration

My data exploration comprised of a variety of techniques, some of which are shown in Figures 1 through 3. An initial

notable discovery was the imbalance in the dataset (Figure 1): a significant majority (77.7%) of the clients are non-defaulters, while a small fraction (22.3%) are defaulters. This imbalance necessitates careful considerations throughout the model development, particularly during the division of data into training and testing sets.

Further, I investigated the presence of outliers in bill statement amounts(BILL_AMT), given the observation of exceptionally high values. Exploration of these features revealed a correlation between high bill statement amounts and high credit limits(LIMIT_BAL). This appears to reflect a proportionate relationship between the credit allocated to clients and their billing amounts (Figure 2), rather than the presence of anomalies. It can then be concluded that the clients with high billing amounts, are the clients that have high credit amounts, therefore are wealthy clients that have considerable spending.

The bar charts in Figure 3 display the distribution of nominal features - Sex, Marriage Status, and Education Level - with respect to default rates. The Sex Distribution chart clearly shows that there is a higher count of females in the dataset; however, the graph also shows that there is no correlation between sex and default rate. This is because the percentage of males defaulting (24.36%) is very close to the percentage of females defaulting (20.97%). The same can be said for the Marriage Distribution chart. The percentages of default are 23.68%, and 21.06% corresponding to Married and Single. It can be concluded that the marriage status also does not have a correlation to the default rate. Lastly, the Education Distribution chart shows potential for a correlation between education level and the default rate. The percentages of default are 19.24%, 23.74%, and 25.30% corresponding to education levels of Graduate School, University, and High School respectively. This clearly shows that as the education level increases, the percentage of default decreases.

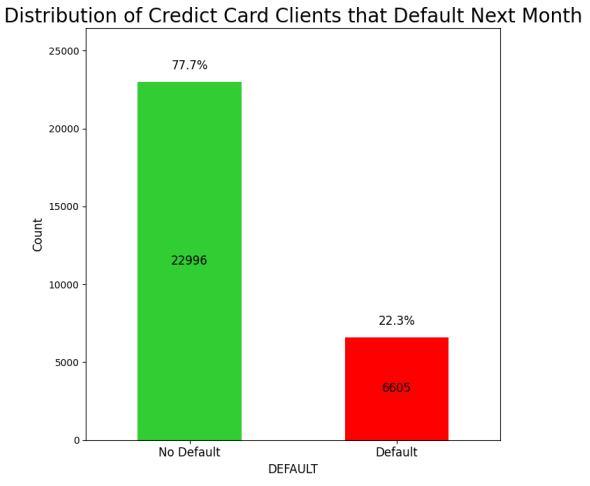

Fig. 2. Proportion of Clients Defaulting versus Non-Defaulting on Credit Payments

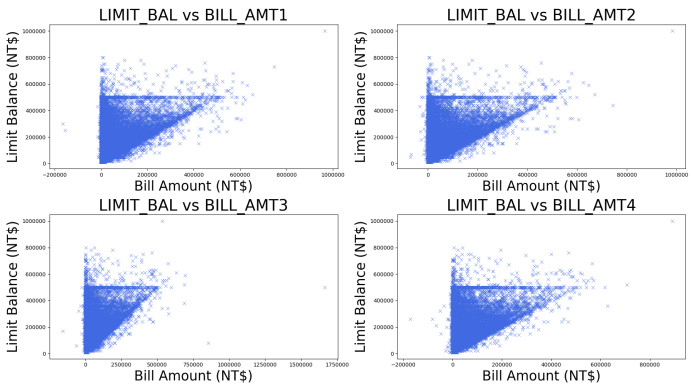

Fig. 3. Outlier Analysis of Monthly Bill Amounts Across All Clients

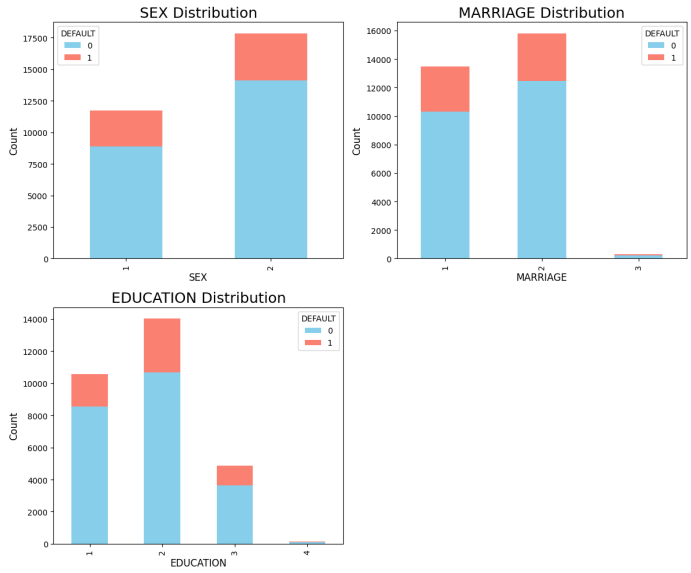

Fig. 4. Distribution of Nominal Features With Respect to Default Rates

2.2 Data Splitting

To split the dataset into training and test sets, the 80-20 split method was used. This meant that 80% of the dataset was allocated to the training set and 20% to the test set. This ratio ensures a balance between having a large training set, while also having a substantial test set to evaluate the performance of the models. Through data exploration, the imbalance of the dataset was discovered. This is an element of the dataset that needs careful considerations. The method I used to address this imbalance is 'Stratified Sampling'. Stratified sampling constructs a split ensuring a consistent proportion of Defaulters and Non-Defaulters in both the training and test sets. TABLE 2 presents key statistics that highlights the effectiveness of the methods used for data splitting.

TABLE 2
Dataset Splitting Description

Statistic	Training Set	Test Set
Count	23680	5981
Defaulters Count	5284	1321
Non-Defaulters Count	18396	4600
Percentage of Default Instances	22.31%	22.31%

2.3 Data Preprocessing

In the development of the machine learning model, various preprocessing techniques were selected to enhance the effectiveness of the training and evaluation stages.

2.3.1 One-Hot Encoding

The first preprocessing method applied to the dataset was One-Hot encoding on nominal features 'SEX', 'MARRIAGE', and 'EDUCATION'. Before implementing one-hot encoding, machine learning models would mistakenly interpret these integers as ordinal, assuming a relative scale. One-Hot encoding avoids this by treating each category as an independent feature, enabling the machine learning model to process the dataset more effectively.

2.3.2 MinMaxScaler

Following one-hot encoding, I used MinMaxScaler to scale the numerical features of the into the range [0,1]. This helps reduce the influence of outliers and improves the convergence of gradient-based optimization algorithms.

2.3.3 Principal Component Analysis (PCA)

After MinMaxScaler, I implemented PCA. PCA is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while retaining a high percentage of variance. This reduction mitigates issues like over-fitting and reducing computational cost. I decided that the dataset should have the first 11 principal components as it captures around 98.33% of the total variance using less than half the original dimensions.

3 Evaluation Metrics

When selecting the most appropriate evaluation metrics for this project, there are a few factors that are critical to acknowledge. These include acknowledgement that the dataset is a binary classification problem and that an imbalance exists between the two binary classifications.

Accuracy evaluates the proportion of true results amongst total cases. This will give an overview into the performance of the model.

Precision measures the ratio of true positives over all positive predictions. This is an important metric since it highlights the cost of false positives.

Recall measures the ratio of true positives to actual positives. This metric will highlight the cost of false negatives (Predicting no default when the client actually defaults).

F1 Score is a combination of precision and recall and is an important metric that highlights the balance both the precision and recall. For each model, I am going to create a confusion matrix as the components in a confusion matrix are used to calculate these different metrics.

Predicted Class	Actual Class	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

4 Model Training

4.1 Algorithm Selection

Baseline Model - Logistic Regression:
I have chosen Logistic Regression as the baseline model. Logistic Regression is a simple and effective model that works effectively on binary classification problems. Further, the preprocessing steps that I have used, such as one-hot encoding make the dataset suitable for Logistic Regression as this model performs well with normalized, linearly separable data.

Proposed Model - Random Forest:
I have chosen Random Forest as the proposed model. Random Forest is known for handling complex, non-linear relationships between features and the target variable. Data exploration in 2.1 revealed that there is a major imbalance in the dataset. Random Forest is particularly effective in such scenarios due to its ensemble nature.

Random Forest's superiority over Logistic Regression:
There are many reasons why Random Forest is considered a better model than Logistic Regression. The 'Default of Credit Card Clients' dataset has 23 feature variables and so over-fitting can prove to be a persistent issue. Random Forest, however, is less prone to over-fitting due to the models ability to generalize. Furthermore, as mentioned before, the dataset is imbalanced and Random Forest builds multiple decision trees on various data samples, ensuring adequate representation.

4.2 Logistic Regression

In the initial stage of my training process, I trained and tested the Logistic Regression model prior to the PCA application. This model served as a basic standard for comparison. To optimize the performance of the model, I evaluated the model's metrics before PCA application, assessing for potentially over-fitting or under-fitting using Learning Curves and Cross-Validation. The analyses concluded that the model was neither over-fitting nor under-fitting. To fine-tune hyperparameters such as maximum number of iterations, I used GridSearchCV. The optimal hyperparameters identified were:
C: 100, Maximum Iterations: 500, Penalty: l1, Solver: liblinear
The selection of a high 'C' suggests a relatively weak regularization, implying that the model benefits from capturing more complexity. The high number of iterations points towards a requirement for extended convergence, possibly due to the complexity of the data. The choice of l1 for penalty may imply that the model benefits from some features being assigned zero coefficients. Finally, the most suited solver could be liblinear due to the binary nature of the target variable.

	Accuracy	Precision	Recall	F1 Score
Before Tuning	0.7769	0.0000	0.0000	0.0000
After Tuning	0.8095	0.6595	0.3020	0.4143

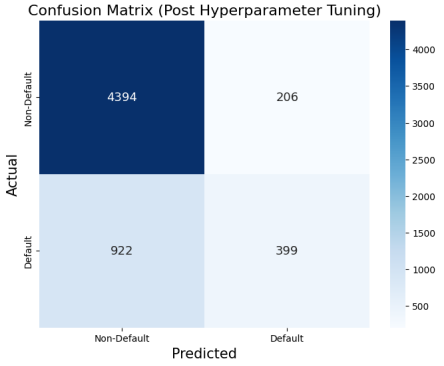

Fig. 5. Logistic Regression Confusion Matrix Post Hyperparameter Tuning

4.3 Random Forest

Development of my proposed model, Random Forest, mimicked a similar approach taken with Logistic Regression. Initially, the Random Forest model was also trained and tested on the dataset split before PCA. This was a standard used for comparison. For hyperparameter tuning, I employed GridSearchCV to identify the most effective parameters such as Max Depth. This method found that the following parameters worked best:
Max Depth: 10, Min Samples Leaf = 4, Min Samples Split = 5, N Estimators = 300
The max-depth of 10 suggests that a medium amount of complexity is needed to capture the essence of the patterns, but going too deep could lead to over-fitting with deeper trees. The min samples leaf at 4 aids in generalization and prevents over-fitting.

	Accuracy	Precision	Recall	F1 Score
Before Tuning	0.8168	0.6612	0.3664	0.4715
After Tuning	0.8183	0.6614	0.3800	0.4827

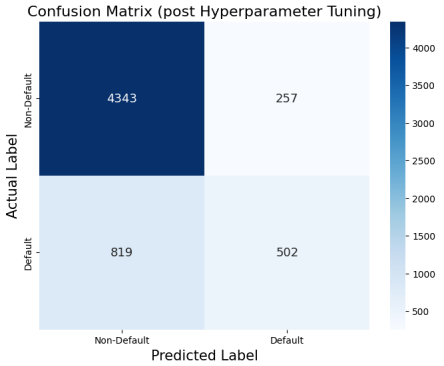

Fig. 6. Logistic Regression Confusion Matrix Post Hyperparameter Tuning

5 Model Comparison

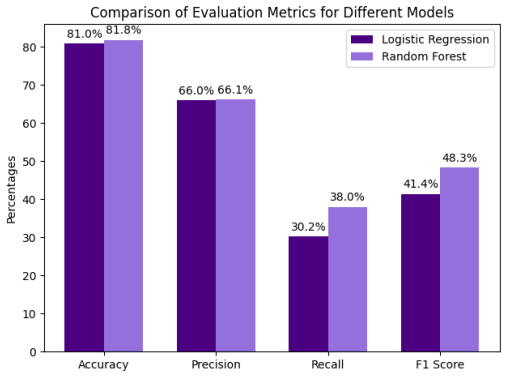

Fig. 7. Logistic Regression Vs Random Forest

The comparison of the baseline model and the proposed model clearly shows that the Random Forest model performs better in all aspects. While accuracy and precision do not get much better, recall improved by around 8% and F1 Score by 7%.

6 Conclusion

Despite achieving a high accuracy and precision, low recall and F1 scores suggest limitations in the current model's ability to predict credit defaults based on historical transactions. A major limitation in this study was the imbalanced nature of the target variable. If I was to work on this project again, I would employ preprocessing techniques such as SMOTE and Cluster Centroids to address this imbalance. I would also explore other models such as K-nearest neighbors and SVM.

7 References

[12]"UCI Machine Learning Repository," archive.ics.uci.edu. https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients