A Comparison of Logistic Regression and Random Forests on Predicting Default of Credit Cards

Aqsa Shabbir

01. Introduction

I aim to solve a binary classification problem to predict whether a credit card client will default on their payment using the Default of Credit Card Clients dataset from the UCI Machine Learning Repository [1]. In the real world, banks must be able to manage risk when they issue credit cards to customers. Two machine learning models, Logistic Regression (LR) and Random Forest (RF), will be applied and evaluated. The goal of the research is to compare these models' performance on structured financial data.

There are numerous research papers that have used various models like Naive Bayes and K-Nearest Neighbor on this dataset [10] [12] . I will aim to compare my results to those obtained by Yeh and Lien (2009), whom analyzed this dataset using Logistic Regression, and further extend the analysis by applying a Random Forest algorithm to the same dataset [11]. Performance metrics such as accuracy, precision, recall, and ROC will be used to evaluate and compare the models.

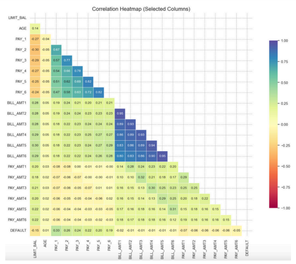

Figure 3: Correlation Matrix

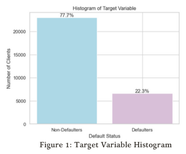

Figure 1: Target Variable Histogram

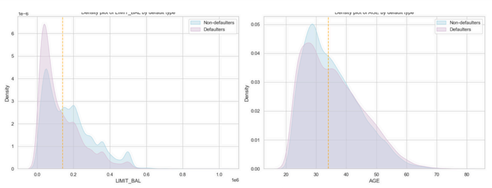

Figure 2: Density Plots

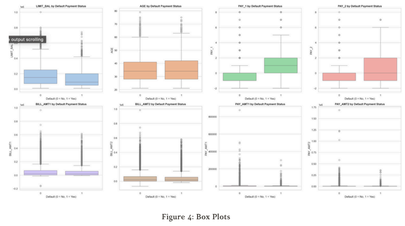

Figure 4: Box Plots

02. Exploratory Analysis

- The dataset consists of 30,000 observations of credit card clients from a bank in Taiwan, including both continuous and categorical variables (such as age, education, history of payments, credit limit, etc). [1]
- **Target Variable**: Default — binary value (1 = default, 0 = no default).
- The dataset does not have any missing values but there are three categorical variables (Education, Marriage, and PAY_n) that are not documented correctly. To address this, I changed the PAY_0 variable to PAY_1 and I added 1 to the values in the column since the minimum was -2 for all of them and should be -1 according to the dataset's documentation. I corrected the EDUCATION column by updating categories 0, 5, and 6, and removed category 0 from the MARRIAGE column since these categories were not documented. [1] Since these are categorical features, I also label encoded them before applying standardization techniques.
- This chart depicts the unbalanced nature of the dataset with around 77% non defaulters and 22% defaulters on their credit card payment. **[Figure 1]**
- **[Figure 2]** The density plots show that defaulters (purple) tend to have lower credit limits and are concentrated in younger age groups compared to non-defaulters (blue), whose credit limits are higher and age distribution skews older. This suggests that individuals with lower credit limits and younger ages may be at higher risk of default, highlighting the importance of considering both demographic and financial factors in credit risk assessment.
- **[Figure 3]** The dataset is multidimensional as it has many attributes. Features that are highly correlated can cause multicollinearity in linear models like LR and reduce interpretability and cause overfitting [2]. I considered dimensionality reduction, but ultimately chose to rely on feature engineering and standardization.
- As can be seen in the box plot graphic **[Figure 4]**, repayment status (e.g., PAY_0, PAY_2) and payment amounts (e.g., PAY_AMT1, PAY_AMT2) emerge as significant features correlated with default, likely due to their direct connection to financial behavior. Other features (e.g., LIMIT_BAL, AGE) show less variation between defaulting and non-defaulting groups, suggesting a lower predictive impact.

03. Hypothesis

Logistic Regression is expected to perform adequately on the given dataset due to its simplicity and ability to handle structured data effectively [2]. Random Forest, on the other hand, is anticipated to achieve higher accuracy and recall, particularly for the minority class, owing to its ensemble nature and ability to capture complex patterns [13]. Previous research has demonstrated the challenges posed by imbalanced datasets in financial credit modeling and highlighted the advantages of ensemble methods like Random Forest in addressing these issues [14].

04. Methodology

1. The dataset was split into a 70:30 ratio for training and testing. The test set was held out and remained unseen until final model evaluation to ensure unbiased performance results.
2. Both training and test sets were standardized using z-score normalization
3. Two models were selected:
   a. Logistic Regression (LR): A simple linear model that is fast to train and computationally efficient.
   b. Random Forest (RF): An ensemble model that performs well on high-dimensional data and captures complex relationships.
4. Hyperparameter Optimization:
   a. Hyperparameters like the lambda (regularization) and solver type were optimized using random search and cross-validation. (LR)
   b. Hyperparameters like the number of trees and minimum leaf size were optimized first using random search, and then grid search was applied to narrow down the best-performing parameters. (RF)
5. 5-fold cross-validation was done on both models during hyperparameter tuning to ensure that results were not overfit to the training data and that generalization performance was evaluated correctly.
6. After hyperparameter optimization, the models were evaluated on the unseen test set using various metrics including accuracy, precision, recall, F1-score, AUC, and log loss.

05. Logistic Regression

1. **Overview**
   a. Logistic Regression is a supervised machine learning algorithm that is used for binary classification problems. When forecasting whether a consumer would default on a loan, for example, it estimates the likelihood that a given input falls into a particular class. The model produces probabilities that are transferred to class labels using a logistic function, often known as a sigmoid function [7].
   b. In terms of mathematics, it is assumed that the input features and the target class's log-odds have a linear relationship. This linear output is converted to probabilities between 0 and 1 using the logistic function. [7]
2. **Pros**
   a. LR is fast to train and is computationally efficient, making it nice for large datasets and real-time applications. [2]
   b. LR performs well on datasets where the relationship between the features and the target is linear. [2]
   c. Unlike other ML models, LR requires little hyperparameter tuning, making it easier and quicker to use. [3]
3. **Cons**
   a. Does not perform that well on non-linear relationships [2]
   b. Outliers can affect it and can skew predictions/insights
   c. Feature engineering is needed to capture nonlinear relationships [9]

06. Random Forests

1. **Overview**
   a. Random forests are an example of a machine learning algorithm that is ensemble and is made up of multiple decision trees. Since it computes the predictions from multiple decision trees that are trained on random subsets of the data (bootstrapping), it reduces overfitting while improving accuracy. This model is robust and works well on complex data that has many features.
   b. There are several hyperparameters (node size, number of trees) that influence the complexity and performance of the model [8].

1. **Pros**
   a. The data used in RFs does not need to be standardized [4].
   b. RF is more robust to overfitting
   c. RF can handle high dimensional data well
2. **Cons**
   a. RF can be computationally intense as it requires more power
   b. RF is harder to interpret than less complicated models like logistic regression
   c. It is robust but can struggle when the data set is imbalanced heavily [6]

07. Experimental Results and Parameter Selection

- For LR, the regularization strength (lambda) and solver typewere tuned. There were a few solvers like lbfgs, sgd, and sparsa that were tested using random search to determine the best combination for minimizing the misclassification rate (MCR). This involved performing 5-fold cross-validation to optimize the hyperparameters and guarantee robustness against overfitting.
- Regularization was used to prevent overfitting, especially since financial data often contains highly correlated features. This multicollinearity can skew the model's coefficient estimates and make the results harder to interpret [2].
- For RF, the number of trees and minimum leaf size were optimized using a random search followed by a grid search to finetune the hyperparameters. These parameters were important for controlling the model's complexity and make sure that the ensemble method did not overfit or underfit the data.
- Since Random forest is an ensemble method, it was predicted to provide a better generalization performance compared to LR, especially on imbalanced datasets. The algorithm's ability to capture non-linear relationships between features and its robustness to outliers made it a good candidate for this problem [6].

Logistic Regression
- I used random search with the range logspace(-4, 0, 10). This corresponds to values: [0.0001, 0.0002, 0.0004, 0.0008, 0.0016, 0.0032, 0.0063, 0.0126, 0.0251, 0.0501].
- The lambda regularization parameter value of 0.0001 was chosen through random search by evaluating the misclassification rate (MCR) across folds.
- The solver was randomly searched over options {'lbfgs', 'sgd', 'sparsa'} and sparsa was selected with the lowest MCR.
- I used 5-fold cross-validation to evaluate misclassification rate for each combination of hyperparameters.

Random Forest:
- For the number of trees, I initially tested values [50, 100, 150] during random search. A value of 150 was chosen after grid search, since it had the best AUC during cross-validation while balancing complexity.
- I also initially tested [1, 5, 10] during random search.
- A larger value like 11 was used since it helps prevent overfitting by making sure that each leaf node represents a sufficiently large subset of the data. This value was selected because it resulted in the highest AUC during cross-validation.
- The final random forest model achieved a test set AUC of 0.7812, outperforming LR and indicating better predictive accuracy.

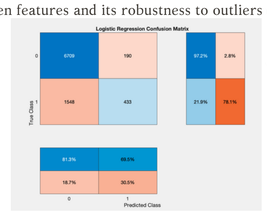

Figure 9: LR Confusion Matrix

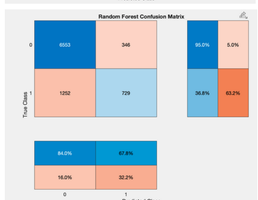

Figure 8: RF Confusion Matrix

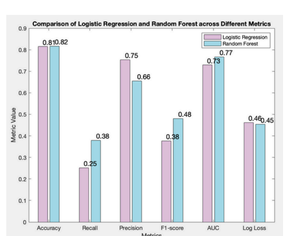

Figure 5: Metrics

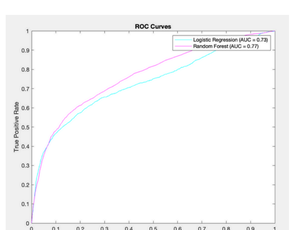

Figure 6: ROC Curves

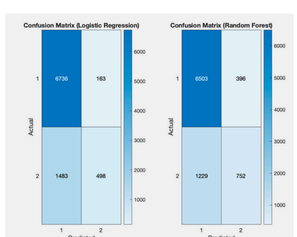

Figure 7: Initial Confusion Matrix

08. Analysis + Evaluation of Results

- To assess whether a model was more accurate in predicting credit card default, the models' performance was examined using a range of evaluation indicators.
- Previous research on this dataset by I-Cheng and Che-hui consists of a more extensive amount of models being compared [12]. The researchers compared models like KNN, logistic regression, and Naive Bayes. The error rate referenced in the research for logistic regression (.20 for the training and .18 for the validation set) is comparable to the .19 training error rate that I computed for the logistic regression model in my evaluation. The research by I-Cheng and Che-hui came to the conclusion that "artificial neural networks should be employed to score clients instead of other data mining techniques, such as logistic regression" [11]. Although, I did not use any ANNs in my work, I came to a similar conclusion about logistic regression and that random forest is a better model than logistic regression for this binary classification problem.

Evaluation Metrics

- **Overall Performance Without Hyperparameter Tuning and Cross-Validation** (*See Figure 7*)
  - Logistic Regression had fewer true positives compared to Random Forest which highlights its struggle with the minority class. It is also important to note that these values were standardized as well. The model showed a significant number of false negatives which demonstrates its limitation in detecting defaulters accurately.
  - It is important to note that Random Forest, even without hyperparameter tuning, showed higher true positives than Logistic Regression.
  - Fewer false negatives show that Random Forest had an advantage in identifying defaulters due to its ability to capture complex feature interactions.
- **Metrics Without Hyperparameter Tuning** (*See Figure 5*)
  - Logistic Regression achieved decent accuracy but was biased toward the majority class.
  - Random Forest had better accuracy, suggesting better performance even without optimization.
  - Logistic Regression's recall was low, indicating a failure to detect defaulters effectively.
  - Random Forest showed better recall, making it more reliable for credit risk applications where identifying defaulters is crucial.
  - Logistic Regression demonstrated acceptable precision but with limited recall, leading to a lower F1-score.
  - Random Forest achieved a better balance of precision and recall, which can be seen by its higher F1-score.
  - Logistic Regression exhibited moderate AUC but was clearly outperformed by Random Forest, which demonstrated higher discriminatory power.
- **Impact of Hyperparameter Tuning and Cross-Validation** (See Figures 8 & 9)
  - With cross-validation, Logistic Regression showed minor improvements in recall and precision, but its limitations with imbalanced data were still seen. It did show a slightly better performance for the majority class.
  - Precision-Recall trade-offs did not significantly shift, maintaining a modest F1-score.
  - Hyperparameter tuning enhanced Random Forest's performance:
    - Higher recall, reducing false negatives further and ensuring a robust model for defaulter detection.
    - Increased AUC (.78) and a steeper ROC curve, demonstrating its improved ability to distinguish between classes.
    - Better precision-recall balance, seen with a higher F1-score.
  - In Logistic regression, there was a marginal decrease in false negatives, indicating minor gains in recall. Also, false positives remained largely unchanged, which emphasizes its inability to handle the class imbalance effectively.
  - In Random Forest, there was a reduced false negatives significantly, affirming its robustness in **detecting defaulters** post-tuning. Increased true positives and reduced false positives reflected the model's adaptability and improved decision boundary through tuning.

- **Log Loss** measures the uncertainty of predictions, and with similar values for both models, Random Forest's slightly lower Log Loss (0.45 vs. 0.47) indicates marginally better overall prediction confidence and reliability, despite it having a lower precision [3].
- Although Random Forest took longer to train **(985 seconds)** due to the ensemble nature and the hyperparameter tuning process, the increase in predictive performance justified the additional computation time. Logistic Regression was much faster **(8.77 seconds)** , but its predictive performance on the imbalanced dataset was limited.

Conclusion

- Random Forest emerged as the better choice for this binary classification problem, particularly due to its superior recall, AUC, and ability to manage imbalanced datasets effectively.
- While Logistic Regression demonstrated computational efficiency and simplicity, it was less suited for scenarios demanding high recall and robust detection of minority classes.
- For credit risk modeling, where accurately identifying defaulters is critical, Random Forest should be preferred, even if it entails higher computational costs. Further enhancements could involve exploring ensemble methods like Gradient Boosting or incorporating artificial neural networks, as suggested by I-Cheng and Che-hui's research [11].

09. Lessons Learned

- It is important to use various different metrics to analyze the performance of models like precision, recall, AUC, etc.
- Class imbalance can have larger implications depending on the model that you are using
- Proper preprocessing and standardization are critical for effective model training and hyperparameter tuning since it impacts model performance.
- Even though Random Forests are computationally expensive, they provide better recall and F1-scores for minority classes.
- Logistic Regression is faster but less effective at capturing complex patterns in data especially when your data is multidimensional and imbalanced.

10. Future Work

- Explore different techniques to handle an imbalanced dataset like SMOTE [6].
- Incorporate more variables that could influence default behavior, such as economic or regional financial indicators, and gather data from multiple years to assess trends and improve the model's generalizability as a whole.
- Extend the analysis by applying and comparing the performance of other more effective machine learning models used in the Yeh and Lien paper, such as: K-Nearest Neighbor, Naïve Bayes, Neural Networks [12].
- Use Recursive Feature Elimination (RFE) to remove the least important features .

References

- [1] I. Yeh, "Default of credit card clients [Dataset]," UCI Machine Learning Repository, 2009. [Online]. Available: https://doi.org/10.24432/C55S3H.
- [2] T. Hastie, R. Tibshirani, and J. Friedman, An Introduction to Statistical Learning: with Applications in R, 2013, p. 114.
- [3] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York: Springer, 2009.
- [4] S. Raschka, Python Machine Learning, 3rd ed. Birmingham, UK: Packt Publishing, 2019.
- [5] I. T. Jolliffe, Principal Component Analysis, Springer Series in Statistics. New York: Springer-Verlag, 2002.
- [6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
- [7] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ: Wiley, 2013.
- [8] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [9] Peng, R. D., Lee, K. H., & Peters, D. (2002). Logistic regression analysis of highly skewed data. Journal of the American Statistical Association, 97(457), 417–427
- [10 Islam, Sheikh Rabiul & Eberle, William & Ghafoor, Sheikh. (2018). Credit Default Mining Using Combined Machine Learning and Heuristic Approach. 10.48550/arXiv.1807.01176.
- [11] Yeh, I., Lien, C., & , (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.
- [12] Dewani, P., Sippy, M., Punjabi, G., & Hatekar, A. (2020). Credit Scoring : A Comparison between Random Forest Classifier and K- Nearest Neighbours for Credit Defaulters Prediction. International Research Journal of Engineering and Technology (IRJET), 07(10), 1887–1892.
- [13] Gang Wang, Jinxing Hao, Jian Ma, Hongbing Jiang, A comparative assessment of ensemble learning for credit scoring, Expert Systems with Applications, Volume 38, Issue 1, 2011, Pages 223-230, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2010.06.048.
- [14] Ibanga, Joseph. (2024). Resolving data imbalance in financial fraud detection by combining machine learning models and ensemble learning strategies. 10.13140/RG.2.2.35330.08644.