Team 56: Boqing (Niko) Zheng, Genna Barge, Kainat Nazir, Yiqiao (Kevin) Wang, Rajat Bajaj

Predictive Modeling for Loan Default Risk: Balance of Return and Risk

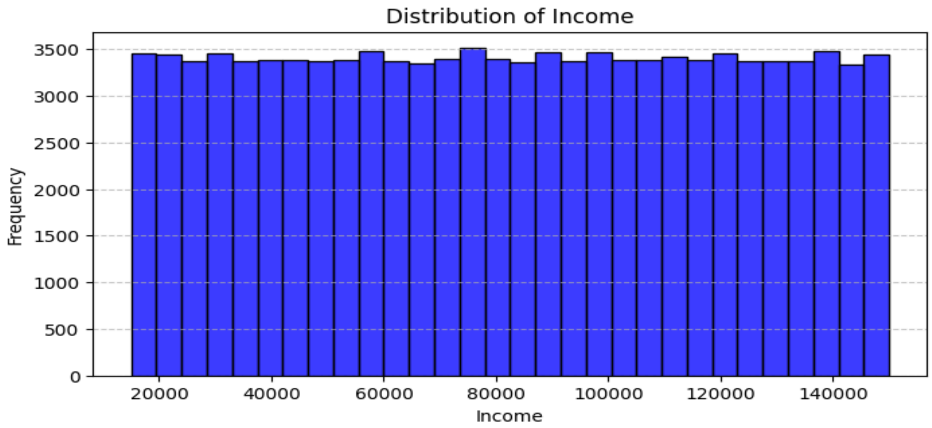
1. Introduction:

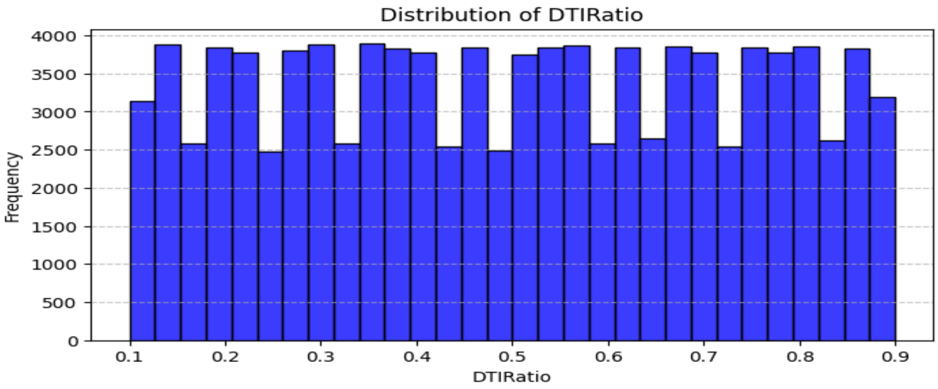
Loan defaults pose a significant challenge for financial institutions, impacting profitability and operational stability. Accurately predicting whether a borrower will default on their loan is crucial for mitigating risks, optimizing lending policies, and improving resource allocation. Deep learning has powerful feature extraction and classification capabilities when faced with massive data and complex features. Therefore, the purpose of this project is to explore how deep learning models can play a role in this field to reduce default risk, while comparing benchmarks of machine learning models commonly used in the industry, such as XGBoost. The dataset has 102,139 records, including demographic, financial, and behavioral attributes such as age, income, loan amount, credit score, and debt-to-income ratio that are critical indicators for assessing default risk, a problem highly relevant in the financial sector, where reducing defaults translates to enhanced profitability and customer retention.

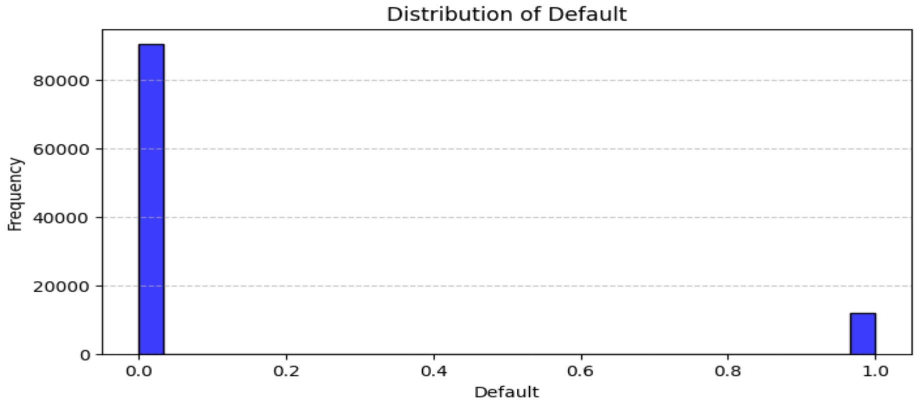
2. Data Preparation:

To prepare the raw data for deep learning, we conducted several processing steps. Missing values were handled by removing incomplete rows to ensure data integrity. Categorical features were converted to numerical representations using one-hot encoding, and numerical features were normalized to prevent gradient issues. The normalization parameters from the training set were applied to the test set to mimic real-world scenarios. We observed significant class imbalance in the target variable “Default,” which was addressed using the SMOTE oversampling technique to balance the category ratio and improve prediction for minority classes. Finally, the processed data was converted to tensor format and efficiently batched using DataLoader for training.

The following three figures show examples of: a variable with a uniform distribution, a variable with a small uneven distribution, and a variable with an uneven distribution.







3. Machine Learning Benchmark

3.1.Modeling

We developed and evaluated multiple machine learning models to predict loan default. The models chosen for benchmarking included Random Forest classifiers and XGBoost classifiers, each tested with three different parameter combinations to optimize their performance. Below, we discuss the configurations and outcomes of these experiments:

- RandomForest1: 100 estimators, maximum depth of 10, min_samples_split of 5, and min_samples_leaf of 2.
- RandomForest2: 200 estimators, maximum depth of 15, min_samples_split of 10, and min_samples_leaf of 4.
- RandomForest3: 300 estimators, maximum depth of 20, min_samples_split of 5, and min_samples_leaf of 2.
- XGBoost1: Learning rate of 0.1, maximum depth of 5, and 100 estimators.
- XGBoost2: Learning rate of 0.05, maximum depth of 7, and 200 estimators.
- XGBoost3: Learning rate of 0.2, maximum depth of 3, and 150 estimators.

3.2. Implementation

We implemented Random Forest and XGBoost models as benchmarks, focusing on hyperparameter tuning to optimize predictive performance. For Random Forest, we defined criteria for tree splitting, tree depth, and estimator numbers, while for XGBoost, parameters like learning rate, tree depth, subsampling, and regularization terms were carefully configured to control overfitting and complexity. Managing class imbalance was a key challenge, particularly for Random Forest, which we addressed with data resampling during preprocessing. Achieving stable performance with XGBoost required experimentation with regularization parameters (reg_lambda and reg_alpha) to balance overfitting control and pattern recognition. Through iterative adjustments, we balanced model performance and training efficiency.

3.3 Results and Evaluation

To assess the predictive power of these models, we focused on three key metrics:

- **AUC**: reflects the overall ability of the model to distinguish between defaulted and non-defaulted customers.
- **F1-Score**: a combined measure of the accuracy and recall of the model's predictions for defaulted customers (focusing on a few categories).

- **Accuracy**: reflects the model's overall rate of correct predictions for all customers (both defaulted and non-defaulted).

The performance of each model was evaluated on the test set using AUC, F1-Score, and Accuracy. The results are summarized below (0.5 threshold):

Model	**Test AUC**	**Test F1-Score**	**Test Accuracy**
RandomForest1	0.673504	0.208579	0.815743
RandomForest2	0.692777	0.195354	0.850793
RandomForest3	0.698694	0.162130	0.861367
XGBoost1	0.719621	0.145049	0.879969
XGBoost2	0.721473	0.158038	0.878990
XGBoost3	0.725191	0.144444	0.879381

The table shows key differences in RandomForest and XGBoost performance for loan default prediction. While RandomForest achieves higher Accuracy, its AUC and F1-Score are lower, with RandomForest3 showing overfitting as tree depth increases. XGBoost outperforms on AUC, indicating better class boundary capture, though its F1-Scores remain low (e.g., 0.1444 for XGBoost3), reflecting challenges in Precision-Recall balance. All models exhibit bias toward the majority class due to test set imbalance, achieving high Accuracy but struggling to identify defaults. Overall, XGBoost3 offers the best complete performance.

4. Deep Learning Attempt

4.1.Modeling

This study employed a Multi-Layer Perceptron (MLP) model and a Deep Residual Network (ResNet) to enhance feature extraction and classification. The MLP comprises five fully connected layers with Batch Normalization to stabilize training, ReLU activation for non-linear representation, and a Sigmoid output layer for binary classification. Its dimensional expansion and compression enable efficient feature screening at different stages.

The ResNet employs multiple Residual Blocks, each containing fully connected layers, Batch Normalization, Leaky ReLU, and Dropout to extract deeper features and mitigate overfitting. Residual connections address gradient vanishing, allowing for deeper networks and improving training efficiency. Projection layers align feature dimensions, ensuring compatibility across layers. He initialization is used to optimize weight training and improve initial performance. This design effectively balances learning ability and stability, significantly improving performance compared to the MLP model.

4.2. Implementation

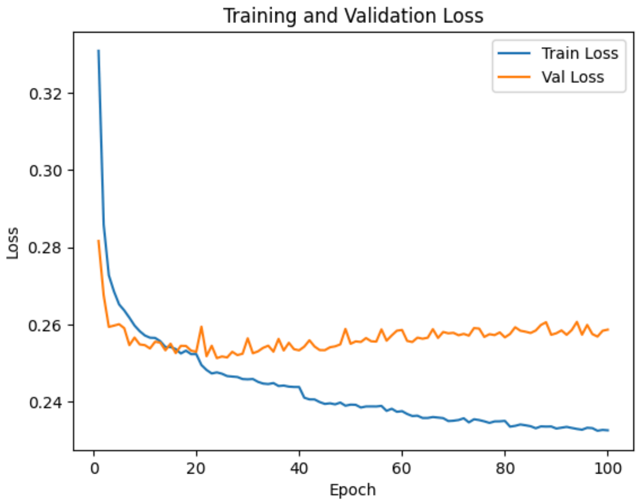
During training, Binary Cross-Entropy Loss was used to measure the difference between predicted results and true labels. The Adam optimizer, with an initial learning rate of 0.001 and weight decay (1×10^{-5}), mitigated overfitting and ensured stable optimization. A learning rate scheduler (StepLR) halved the learning rate every 20 epochs to refine performance in later training stages. For the MLP, the model trained for 100 epochs, leveraging carefully selected hyperparameters. Similarly, for ResNet, extensive hyperparameter tuning was conducted, exploring various dropout rates, regularization parameters, and layer dimensionalities. After each training iteration, models were evaluated on the test set using AUC, F1-Score, and Accuracy to assess performance.

The primary challenges included overfitting, optimization difficulties in deep networks, and threshold selection. Overfitting was addressed through weight decay (L2 regularization) and

Dropout, improving the model's generalization ability. To enhance deep network optimization, the learning rate scheduler and ResNet's residual connections mitigated gradient vanishing and ensured stable convergence. A comprehensive hyperparameter search further refined the configuration, guided by validation loss, AUC, and other metrics. Finally, performance under different classification thresholds was visualized and optimized to ensure the model's robustness and practical application.

4.3 Results and evaluation

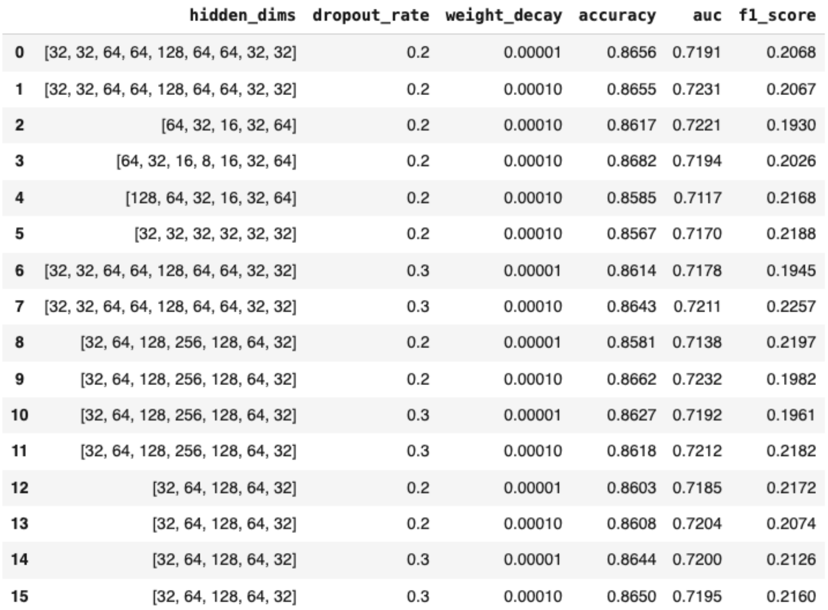
For MLP, we trained as shown below.



From the training curve, it can be seen that the model's training loss and validation loss decrease rapidly within the first 10 rounds, indicating that the model effectively learns the data features; in the middle (10-50 rounds), the training loss continues to decrease, the validation loss tends to stabilize, and the model's performance gradually converges; in the late stage (50-100 rounds), the training loss continues to decrease while the validation loss fluctuates slightly or even rises slightly, indicating that the model may have a certain degree of This suggests that the model may have a certain degree of overfitting. This suggests that we can consider increasing regularization, using early stopping strategy or expanding the dataset to further optimize the generalization ability of the model.

On the test set, MLP achieved Test Accuracy: 0.8595, Test AUC: 0.7174, Test F1-score: 0.2281. Compared to XGBoost, our MLP outperforms it only in terms of its F1-score. This means that we need to optimize further from multiple aspects of the model, which makes our DeepResNet a necessary attempt.

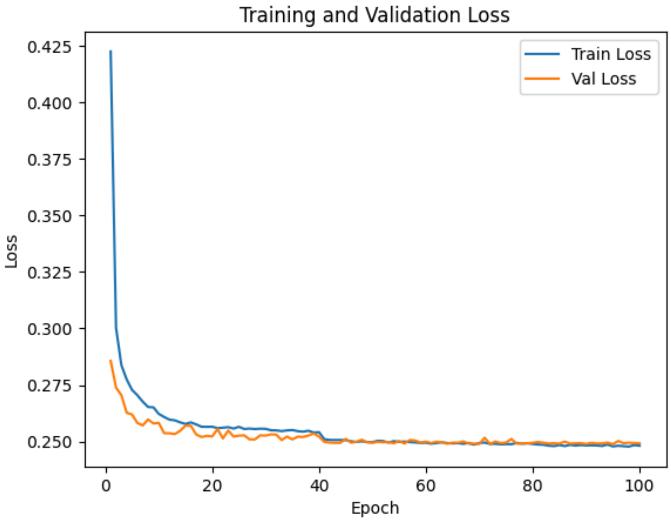
For DeepResNet, we tried the following major model configurations to explore the effects of hidden_dims, dropout_rate, and weight_decay on model performance:



	hidden_dims	dropout_rate	weight_decay	accuracy	auc	f1_score
0	[32, 32, 64, 64, 128, 64, 64, 32, 32]	0.2	0.00001	0.8656	0.7191	0.2068
1	[32, 32, 64, 64, 128, 64, 64, 32, 32]	0.2	0.00010	0.8655	0.7231	0.2067
2	[64, 32, 16, 32, 64]	0.2	0.00010	0.8617	0.7221	0.1930
3	[64, 32, 16, 8, 16, 32, 64]	0.2	0.00010	0.8682	0.7194	0.2026
4	[128, 64, 32, 16, 32, 64]	0.2	0.00010	0.8585	0.7117	0.2168
5	[32, 32, 32, 32, 32, 32]	0.2	0.00010	0.8567	0.7170	0.2188
6	[32, 32, 64, 64, 128, 64, 64, 32, 32]	0.3	0.00001	0.8614	0.7178	0.1945
7	[32, 32, 64, 64, 128, 64, 64, 32, 32]	0.3	0.00010	0.8643	0.7211	0.2257
8	[32, 64, 128, 256, 128, 64, 32]	0.2	0.00001	0.8581	0.7138	0.2197
9	[32, 64, 128, 256, 128, 64, 32]	0.2	0.00010	0.8662	0.7232	0.1982
10	[32, 64, 128, 256, 128, 64, 32]	0.3	0.00001	0.8627	0.7192	0.1961
11	[32, 64, 128, 256, 128, 64, 32]	0.3	0.00010	0.8618	0.7212	0.2182
12	[32, 64, 128, 64, 32]	0.2	0.00001	0.8603	0.7185	0.2172
13	[32, 64, 128, 64, 32]	0.2	0.00010	0.8608	0.7204	0.2074
14	[32, 64, 128, 64, 32]	0.3	0.00001	0.8644	0.7200	0.2126
15	[32, 64, 128, 64, 32]	0.3	0.00010	0.8650	0.7195	0.2160

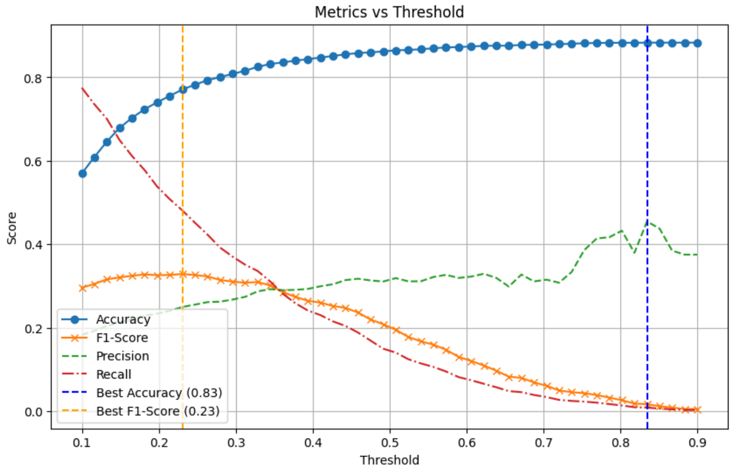
For the loan default prediction task, we chose the model configuration hidden_dims=[32, 64, 128, 256, 128, 64, 32], dropout_rate=0.2, and weight_decay=0.0001, which exhibits the highest AUC on the test set (0.7232) along with a high accuracy (0.8662) and applicable F1 score (0.1982). The high AUC ensures the model's ability to differentiate between defaulting and non-defaulting customers, while the good accuracy reflects the overall predictive stability. This model adopts a deep structure that helps to capture the complex relationship of loan default characteristics, while overfitting is avoided through reasonable regularization parameters and generalization ability is improved. Taken together, the model can effectively balance business requirements and is suitable for real default risk prediction scenarios. Compared with the Benchmark, we find that compared to MLP, DeepResNet is superior in Accuracy and AUC, but the F1-score is slightly decreased. Compared to XGBoost, DeepResNet has a better F1-score with basically equal Accuracy and AUC, and is more adept at distinguishing defaults.

The training plot of this model is:



From the training and validation loss curves, it can be seen that the model decreases rapidly within the first 10 rounds, indicating that the model effectively learns the data features; during the period of 10-50 rounds, the training loss continues to decrease slowly, the validation loss tends to stabilize, and the model gradually reaches a better convergence state; in the later period (50-100 rounds), the training loss decreases slightly, while the validation loss stays stable, and there is no obvious overfitting phenomenon. This indicates that the model training process is smooth, with good generalization ability, and the current parameter settings are more reasonable.

Next, we use the optimal DeepResNet model to compare its Accuracy and F1 scores using different thresholds:



This figure shows the variation of the model's performance metrics under different thresholds, including Accuracy, F1 score, Precision, and Recall. From the figure, it can be seen that as the thresholds change, Accuracy reaches its highest at high thresholds (0.83), while F1 score reaches its best at lower thresholds (0.23), indicating that there is a trade-off between Precision and Recall. At high thresholds, precision is higher but recall decreases, implying that the model is more “conservative” and tends to predict fewer defaulting customers. At low thresholds, recall increases but precision decreases, meaning that the model is more “conservative” and tends to capture all possible defaulting customers. If the business is more concerned with reducing default risk and needs to ensure a high recall rate, the threshold corresponding to the best F1 score (yellow dashed line) can be selected. If the business is more concerned with overall prediction accuracy, the threshold corresponding to the best accuracy (blue dashed line) can be selected..

5. Deployment:

Ultimately, we propose a flexible multi-stage screening scheme: using low-threshold models to expand risk coverage in the initial screening stage, and adopting high-threshold models to ensure accuracy in the in-depth review stage. This dynamic threshold adjustment strategy, combined with powerful deep learning model capabilities, provides commercial organizations with an intelligent solution to balance efficiency and risk in different business scenarios.

The deep learning model will be integrated into the institution’s loan processing system to enable real-time prediction of default probabilities. Key applications include loan application screening, where high-risk applicants are flagged for review and low-risk ones benefit from faster approvals, automated loan approval assistance based on risk thresholds, and fraud detection by identifying anomalies in data patterns. Regular retraining is essential to prevent model drift and maintain accuracy over time. Ensuring data privacy and security, such as through encryption and compliance with regulations, is critical, as is updating IT infrastructure to support integration.

Ethical considerations must also be addressed, including transparency to inform customers about AI involvement in decisions, auditing the model to prevent bias against demographic groups, and establishing accountability to ensure alignment with ethical and legal standards. These measures ensure a robust, fair, and secure deployment of the model.

6. Conclusion

This project explores the application of deep learning models (e.g., DeepResNet) and traditional machine learning models (e.g., XGBoost, Random Forest) in loan default prediction, which, combined with a dynamic threshold adjustment strategy, provides financial institutions with a flexible risk management solution. Experiments show that XGBoost excels in overall accuracy and efficiency, and is suitable for rapid approval scenarios for low-risk customers, while DeepResNet performs better in complex feature extraction and default customer identification, and is especially suitable for high-risk prevention and control. Through a multi-stage screening strategy with low thresholds to expand risk coverage and high thresholds to improve audit accuracy, the model can effectively balance recall and precision, optimize the loan approval process, reduce default risk and improve resource utilization efficiency. This solution combines model performance with business requirements, demonstrating the potential of data-driven intelligent decision-making in the financial industry.

Bibliography

Opa, Valentine Ojong, and Wendy Tabe-Ebob. *The Effects of Loan Default on Commercial Banks Profitability: Case Study BICEC Limbe*. 21 Dec. 2020.

Provost, Foster, and Tom Fawcett. *Data Science for Business*. O'Reilly Media, Inc., 2013.

"What Happens if You Default on a Business Loan?" *Bankrate*, 21 Sept. 2023, www.bankrate.com/loans/small-business/what-if-you-default-on-business-loan/?tpt=a. Accessed 13 Oct. 2024.

Data

Nikhil. "Loan Default Prediction Dataset." *Kaggle*, 1 Oct. 2022, www.kaggle.com/datasets/nikhil1e9/loan-default.