Predicting Loan Applications from Randomized Control Trial Data

Nuha Pagarkar, Mithat Kus, Nandini Sistla, Anbo Wang, Xinrong Hao

Columbia University, Data Analytics Project

Abstract

We analyze a randomized control trial dataset of 53,000 microcredit customers to predict loan application behavior based on advertising features and customer characteristics. Despite testing seven machine learning models, the best performance achieves ROC-AUC of only 0.70, which revealed fundamental prediction challenges: severe class imbalance (91.5% non-applicants), weak feature correlations, and complex non-linear interactions. Key predictors include customer dormancy, interest rates, and transaction history. This project demonstrates the difficulty in predicting financial decisions and provides actionable insights for targeted marketing in microcredit lending.

1 Data and Preprocessing

1.1 Dataset Overview

The dataset contains 58,168 observations with 37 features, reduced to 53,194 after removing missing values in experimental features. The target variable `applied` is binary (0=no application, 1=application).

Features include:

- *Demographics*: race, gender (female), education (edhi), risk category
- *Behavior*: dormancy (months since last loan), transaction count (trcount), previous borrowing
- *Experimental treatments*: interest rate (offer4), prize incentive, deadline length, photo types, demographic matching

1.2 Data Cleaning

Leakage prevention: We removed post-application variables: `tookup` (loan acceptance), `badacct_last` (defaults), `amountbrw_unc` (amount borrowed), and temporal behaviors measured after application decisions.

Missing Value Handling: Instances with null values in the "prize" column were removed from the dataset.

Multicollinearity: VIF analysis revealed `deadlinemed` had VIF=18.81 due to experimental design constraints, so it was dropped.

Encoding: Tree-based models used label encoding for categorical variables (race, risk); logistic regression used one-hot encoding with dropped baselines to avoid perfect multicollinearity.

Final dataset: 53,194 observations, 27 features.

2 Exploratory Data Analysis

2.1 Severe Class Imbalance

The target variable shows extreme imbalance:

- Did not apply: 48,672 (91.5%)
- Applied: 4,522 (8.5%)

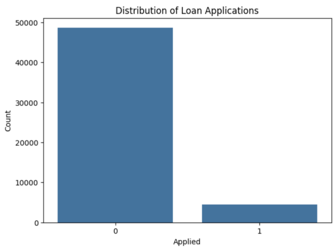

Figure 1: Distribution of Application

This 11:1 ratio presents a fundamental modeling challenge and reflects real-world conversion difficulty. To deal with the imbalance, we considered random undersampling and oversampling. Since oversampling would require around 45 thousand rows of synthetic data, it would alter our original dataset severely. We believed that random undersampling would be more suitable for our data and decided to test it on logistic regression. If we get better results with random undersampling, we decided we could change our dataset accordingly.

2.2 Weak Correlations

Correlation analysis reveals only two features have correlation >0.15 with `applied`, which were dormancy (0.17), and risk (0.15). This suggests application decisions emerge from complex, non-linear feature interactions rather than simple linear relationships.

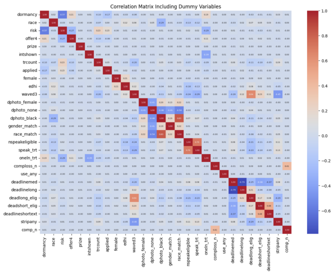

Figure 2: Correlation Heatmap

2.3 Customer Characteristics

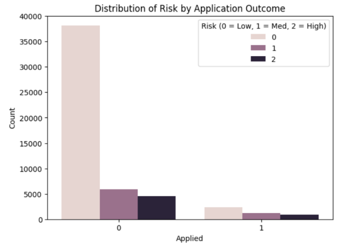

Figure 3: Dist. of Risk by Application Outcome

The distribution of risk levels indicates that low-risk customers constitute the largest segment of the sample, followed by medium- and high-risk groups. This pattern is consistent across both applicants and non-applicants. Low-risk customers also account for the largest number of applications in absolute terms, reflecting their greater representation in the dataset.

Analysis of dormancy across application outcomes shows that non-applicants (class 0) and applicants (class 1) are highly concentrated at the lowest dormancy values, suggesting that recent customer activity alone does not guarantee application behavior.

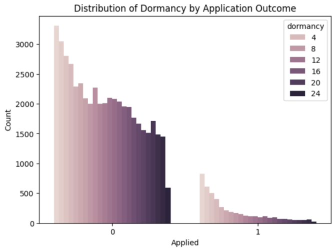

Figure 4: Dist. of Dormancy by Application Outcome

This can indicate that advertisements are disproportionately viewed and not acted upon by customers who have been active in loan take-up. This would then back up why a large majority of the response group is non-applicants.

3 Modeling Methodology

We evaluate seven algorithms with 80-20 train-test splits, stratified by target to preserve class balance. All models handle imbalance via class weighting except where noted.

3.1 Logistic Regression (sklearn)

We applied StandardScaler preprocessing, balanced the class weights to address class imbalance, and employed L2 regularization.

Results: After manually adjusting the threshold (0.63) to maximize F1 score, we obtained F1-score: 0.261, recall 0.4, precision: 0.192. The model recovers a meaningful share of applicants but at the cost of many false positives, reflected in the low precision.

Insights: The F1-score for customers who did not apply (0.78) is much higher than consumers who applied (0.24). This shows us that the model is having a harder time identifying consumers who applied, which is the underrepresented class. This is a clear sign of class imbalance impacting the model negatively.

3.2 Logistic Regression (statsmodel)

Since we want to understand what features of ads and customer characteristics influence application decisions, we ran a statsmodel logistic regression to analyze the coefficients.

Statistical significance ($p < 0.05$): Negative effects from dormancy (-0.071), interest rate (-0.045), waved3 (-0.289); positive effects from education (0.107), oneln_trt (0.111), long deadlines (0.170), LOW risk (0.517), MEDIUM risk (0.597).

Insights: Longer dormancy and higher rates deter applications; lower-risk and educated customers apply more; long deadlines outperform short ones.

3.3 Class Imbalance Handling

We also tested random undersampling (balancing to 4,522 samples each class). The resulting ROC-AUC was 0.688, which is worse than class weighting on full data. Therefore we will disregard undersampling as an option. Class weighting is a better method; undersampling discards valuable information.

3.4 K-Nearest Neighbors

Distance-based method with StandardScaler, distance weighting, hyperparameter search over $k \in [180, 200]$.

Results: Best $k = 192$ (very large), ROC-AUC=0.670

Analysis: Optimal k covering ∼1% of training data indicates extreme sparsity in 27-dimensional space. KNN suffers from curse of dimensionality and is unsuitable for this problem.

3.5 Random Forest

Ensemble of 300 trees, balanced class weights, RandomizedSearchCV tuning.

Best params: n_estimators=200, min_samples_leaf=8, min_samples_split=2, max_features=0.5

Results: ROC-AUC=0.683, Accuracy=0.86

Feature importance (top 3): dormancy, offer4, trcount. This confirms customer history and pricing drive decisions more than experimental treatments.

3.6 XGBoost

Gradient boosting with scale_pos_weight=10.76 for imbalance, learning_rate=0.05, max_depth=4.

Best params: n_estimators=200, learning_rate=0.1, max_depth=5

Results: ROC-AUC=**0.701** (best), Precision(1)=0.16, Recall(1)=0.62

XGBoost achieves best discrimination, capturing non-linear patterns while matching logistic regression's AUC.

Feature importance (top 3): Risk, dormancy, intshown

3.7 LightGBM

Leaf-wise gradient boosting, computationally efficient alternative to XGBoost.

Best params: num_leaves=15, n_estimators=200, max_depth=3, learning_rate=0.01

Results: ROC-AUC=**0.701** (tied), Precision(1)=0.16, Recall(1)=0.62

Feature importance (top 3): dormancy, offer4, trcount.

LightGBM matches XGBoost performance with faster training, confirming both have reached the dataset's performance ceiling.

3.8 Dense Neural Networks

Systematic architecture exploration:

Hidden layers (64 nodes, ReLU): 1 layer=65.04% (best), 2=63.74%, 5=61.85%. More layers decrease performance—data doesn't benefit from deep hierarchies.

Nodes per layer (2 layers, ReLU): 4 nodes=67.32% (best). Very small networks optimal, indicating low-dimensional representations.

Activations (2 layers, 32 nodes): Sigmoid=66.36% (best), ReLU=63.75%.

Final architecture: 2 hidden layers (32 nodes each), ReLU activation, sigmoid output, Adam optimizer, balanced class weights.

Results: ROC-AUC=0.692, stable training without overfitting, but no improvement over gradient boosting. Complex models do not help. The problem is not the model, rather the lack of signal within the data.

4 Results and Comparison

Table 1: Model Performance Summary

Model	**AUC**	**Prec**	**Rec**
Logistic Reg.	0.703	0.15	0.64
Logistic (Bal.)	0.688	0.65	0.63
KNN	0.670	0.33	0.00
Random Forest	0.683	0.20	0.21
XGBoost	**0.701**	0.16	0.62
LightGBM	**0.701**	0.16	0.62
Neural Net	0.692	0.23	0.11

Best models: Logistic Regression, XGBoost and LightGBM (AUC=0.70). Logistic regression matches XGBoost and LightGBM and offers better interpretability.

Conclusion: Since interpretability is very critical for loan-take up analysis, we will consider logistic regression as our best model.

4.1 Performance Interpretation

ROC-AUC of 0.70 indicates fair discrimination performance(0.5=random, 0.7=fair, 0.8=good, 0.9+=excellent). Models perform significantly better than random but fall short of strong predictive power.

Why only 0.70 AUC?

1. **Inherent unpredictability**: Human financial decisions involve unmeasured factors (personal circumstances, competing offers, psychology).
2. **Weak treatment effects**: Experimental variations show modest impact as no treatment dramatically changes behavior.
3. **Complex interactions**: Weak linear correlations suggest decisions emerge from complex, non-linear combinations.
4. **Imbalance challenges**: With 91.5% non-applicants, models must be extremely confident to predict applications.

4.2 Feature Importance Synthesis

Consistently across models, top 4 features are:

1. **dormancy**: Longer dormancy → lower application probability
2. **offer4** (interest rate): Higher rates deter applications
3. **trcount**: More transactions → higher engagement → more applications
4. **risk**: Lower risk → higher application rates

5 Business Implications

5.1 Targeting Strategy

Prioritize customers with: (1) Low dormancy (<9 months), (2) High transaction count (≥5 loans), (3) Low/medium risk, (4) Competitive rates (<8%). The results from our logistic regression model support this outcome as all of these features are statistically significant.

Strategy: Use XGBoost scores to rank customers monthly and target top 20-30% most likely to apply. Based on our model's results, we would expect higher conversion rates, lower costs, personalized offers.

5.2 Limitations

Modest power: AUC=0.70 means ∼30% of the time our models ranks non-applier higher than applier—limits improvement magnitude.

Threshold selection: With severe class imbalance, performance is highly sensitive to the classification threshold. Higher thresholds ($\approx$ 0.3–0.4) are appropriate when outreach is costly, while lower thresholds ($\approx$ 0.1–0.2) are appropriate when missing potential applicants is more costly. Thus, threshold choice should be guided by business costs rather than purely statistical criteria.

External validity: Model trained on historical RCT data may degrade if population, market, or products change. The model would need to be updated regularly.

6 Conclusion

This project evaluates machine learning approaches for predicting loan application behavior using large-scale RCT data from a microcredit lender. Despite extensive modeling across seven algorithms, performance plateaus at ROC-AUC ≈0.70. This ceiling reflects fundamental limitations of our data: severe class imbalance, weak marginal effects of experimental treatments, and substantial unobserved drivers of individual financial decisions.

Across models, customer history consistently dominates predictive power. Dormancy, transaction count, risk category, and interest rates are the strongest predictors, and most advertising treatments exhibit little effects. Gradient boosting methods modestly outperform linear and distance-based models by capturing non-linear interactions, but gains remain limited. Neural networks offer no additional benefit, which indicates that the problem is data, rather than model-constrained. From a practical perspective, logistic regression provides a competitive and interpretable baseline, while gradient boosting is best suited for ranking customers in targeting applications.

References

[1] M. Bertrand, D. Karlan, S. Mullainathan, E. Shafir, J. Zinman, “What’s Advertising Content Worth? Evidence from a Consumer Credit Marketing Field Experiment,” *Quarterly Journal of Economics*, vol. 125, no. 1, pp. 263–306, 2010.

[2] A. Banerjee et al., “The Miracle of Microfinance? Evidence from a Randomized Evaluation,” *American Economic Journal: Applied Economics*, vol. 7, no. 1, pp. 22–53, 2015.

[3] T. Fawcett, “An Introduction to ROC Analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[4] J. Kleinberg, J. Ludwig, S. Mullainathan, Z. Obermeyer, “Prediction Policy Problems,” *American Economic Review Papers & Proceedings*, vol. 105, no. 5, pp. 491–495, 2015.

[5] T. Chen, C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[6] H. He, E. A. Garcia, “Learning from Imbalanced Data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.