

Dynamic Pricing and Risk-Based Loan Offers

Vedant Mohadikar¹, Mardav Jain², Ziyang Sheikh³, Bhavni Singh⁴, Prof. Prajwalita Dongre⁵

^{1,2,3,4,5}

MIT Art, Design & Technology University, Pune

Abstract—This paper presents a student-oriented, implementable framework for Dynamic Pricing and Risk-Based Loan Offers aimed at digital lenders and fintech platforms. The approach fuses borrower-level credit risk assessment with market-aware pricing and constrained optimization to produce individualized loan terms in near real time. We expand on data curation and feature engineering, hybrid risk modeling (scorecards + gradient-boosted trees), acceptance-elasticity pricing, constrained optimization, and operational deployment notes. Simulated experiments demonstrate improved net interest margin (NIM) and acceptance rates while keeping portfolio PD within predefined limits. Emphasis is placed on explainability, fairness monitoring, and iterative feedback loops for safe adaptation in production environments.

Index Terms—Dynamic Pricing, Risk-Based Lending, Credit Scoring, Machine Learning, Optimization, Explainable AI, Fin-Tech

I. INTRODUCTION

The lending industry is rapidly evolving as digital channels and data-driven decision engines replace manual underwriting processes. Lenders must balance two competing objectives: provide competitive pricing to attract customers, and protect portfolio integrity by limiting credit losses. Static, tiered-pricing models — which assign broad categories and fixed rates — are often inefficient; they can overcharge creditworthy consumers while failing to optimize for borrower heterogeneity or market dynamics.

Dynamic pricing adapts offers to borrower characteristics and market conditions in (near) real time. When coupled with risk-based pricing—where interest rates reflect estimated default probabilities—lenders can improve profitability and fairness simultaneously. However, practical deployment faces multiple operational challenges: explainability for regulators and customers, sub-second response times

for user experience, fairness checks to avoid discriminatory patterns, and robust feedback loops to handle data drift and changing market regimes.

This paper proposes a pragmatic, reproducible framework intended for student projects and early-stage fintech proto-types. We emphasize an end-to-end pipeline: data ingestion, feature engineering, hybrid risk modeling, acceptance estimation, constrained optimization, offer composition, and monitoring. Our presentation blends technical detail (equations, algorithms) with clear explanatory prose to be both rigorous and accessible.

II. PROBLEM STATEMENT

We formalize the problem as follows. A stream of loan applications indexed by i arrives at time t providing applicant features X_i (demographics, credit history, behavioral signals) and market context M_t (base rates, competitor pricing, macro indicators). For each application, the system must produce an offer $O_{i,t} = (r_{i,t}, L_{i,t}, T_{i,t})$ — interest rate, loan amount, tenure — that maximizes expected institutional utility while meeting compliance and risk constraints. We consider the single-offer utility:

$$U(r) = p_{\text{acc}}(r | X_i, M_t) \cdot E[\text{Profit} | r, X_i] = p_{\text{acc}}(r)(1 - PD(X_i)) \Pi(r, L)$$

where:

- $PD(X_i)$ is the probability of default predicted for borrower i .
- $p_{\text{acc}}(r | X_i, M_t)$ is the probability the borrower accepts an offer at rate r .
- $\Pi(r, L)$ denotes expected discounted profit at rate r and exposure L , incorporating expected default losses (LGD).

Optimization occurs under constraints:

$$PD(X_i) \leq PD_{max}, \quad \text{Fairness metrics} \leq \delta, \quad \sum_{i \in S} L_{i,t} \leq E_{max}$$

for portfolio slices S . The system must be auditable: each automated decision must record model version, input snapshot, and explanation.

III. LITERATURE REVIEW

We expand the literature review into focused subsections to position our approach and draw practical insights.

A. Credit Scoring and PD Estimation

Traditional credit scoring relies on logistic regression and scorecards, favored for transparency and regulatory acceptance [1]. Scorecards transform features into weight-of-evidence (WoE) bins and combine them into an additive logistic model; they remain the foundation for many underwriting pipelines. Modern enhancements layer tree-based ensembles (XGBoost, LightGBM) to capture nonlinear interactions and boost discrimination (AUC). Ensemble methods typically outperform simple scorecards in predictive accuracy, but require XAI tools (SHAP, LIME) to regain interpretability [2], [4].

B. Dynamic Pricing Approaches

In dynamic pricing literature, approaches vary from econometric demand modeling to machine learning and reinforcement learning (RL) frameworks. Contextual bandits and RL algorithms optimize price taking into account long-term effects (e.g., customer lifetime value) while balancing exploration and exploitation [3]. However, in lending, exploration (testing higher rates) is constrained by regulatory risk limits and potential customer harm. Hence, many practical systems prefer constrained optimization on historical elasticity estimates rather than fully exploratory RL.

C. Acceptance Modeling and Elasticity

Modeling the acceptance probability $p_{acc}(r)$ is central. Empirical work models' acceptance as a decreasing function of interest rate, often via logistic regression; additional covariates (income, prior defaults, tenure) capture heterogeneity in elasticity. Randomized A/B

trials and field experiments are reliable ways to estimate elasticity in production, while observational methods (propensity scoring) can be used on historical offer-response data when experiments are unavailable.

D. Optimization under Risk and Fairness Constraints

Optimization in finance frequently employs constrained objective functions that balance expected return and risk (e.g., mean-variance, CVaR). In our context, constraints include per-applicant PD limits, portfolio exposure ceilings, and disparity thresholds to limit unfair outcomes across protected groups. Recent research proposes integrating fairness constraints directly into optimization (constrained maximization) or using post-processing to adjust outputs to satisfy fairness criteria.

E. Explainable AI, Governance and Deployment

Explainability is crucial in lending due to regulatory requirements. Techniques like SHAP decompose model outputs into feature attributions, enabling concise explanations for individual decisions. Governance frameworks require model versioning, audit logs, performance monitoring, and pre-deployment bias testing (disparate impact analysis). Our architecture adopts these best practices and emphasizes human in-the-loop escalation for borderline or high-risk cases.

F. Gap and Contribution

While prior studies address components of this pipeline, few provide an end-to-end, operationally realistic blueprint that combines hybrid risk modeling, acceptance elasticity, constrained optimization, explainability, and deployment considerations in a student-accessible manner. Our contribution is a reproducible, implementation-forward framework that documents both design rationale and practical trade-offs.

IV. DATA PREPARATION

Data is the backbone. We describe recommended datasets, preprocessing strategies, and feature design in detail.

A. Data Sources and Privacy

Combining internal origination data (applicant attributes, decisions, payment history) with external inputs (credit bureau, macro indicators, competitor rate snapshots) yields rich modeling inputs. Student projects should ensure PII is masked and synthetic augmentation is used when real data is unavailable.

B. Feature Engineering

Key engineered features:

- Behavioral: rolling payment timeliness, frequency of partial payments, variance in payment timing.
- Credit usage: utilization ratios, changes in limit-to-balance percentage, recent increases in balances.
- Temporal: time since last delinquency, seasonality indicators (month, festival/sale flags).
- Market: short-term spread between base rate and competitor median, volatility indices.

Feature interactions (e.g., income × utilization) are included to capture nonlinear risk drivers.

C. Handling Imbalanced Outcomes

Defaults are rare in many datasets, so techniques like stratified sampling, SMOTE for synthetic minority oversampling, or cost-sensitive learning help models learn from minority classes. However, avoid naive oversampling that corrupts temporal dependencies.

D. Train/Test Splits and Temporal Validation

Avoid random splits; use time-based splits to mimic production deployment (train on older windows, validate on recent windows). This reduces optimistic leakage and helps detect data drift when tested on later periods.

V. METHODOLOGY AND FRAMEWORK

We present a detailed methodology combining models and optimization, mixing equations with pragmatic steps.

A. Hybrid Risk Modeling

We combine a logistic scorecard and an ensemble:

- a) *Scorecard baseline:* A scorecard computes log-odds:

$$\log \frac{P(D = 1 | X)}{1 - P(D = 1 | X)} = \alpha_0 + \sum_j \beta_j \cdot \text{WoE}(X_j)$$

Where,

WoE denotes weight-of-evidence for bins of feature X_j . Scorecards are calibrated for interpretability and regulatory acceptance.

- b) *Ensemble model:* An XGBoost model provides higher discrimination:

$$P^{\wedge} D_{xgb} = \sigma(f_{xgb}(X))$$

We blend outputs via convex combination:

$$PD_{\text{hybrid}} = \lambda \cdot P^{\wedge} D_{\text{scorecard}} + (1 - \lambda) \cdot P^{\wedge} D_{xgb}$$

λ controls the interpretability-accuracy trade-off and is tuned via validation.

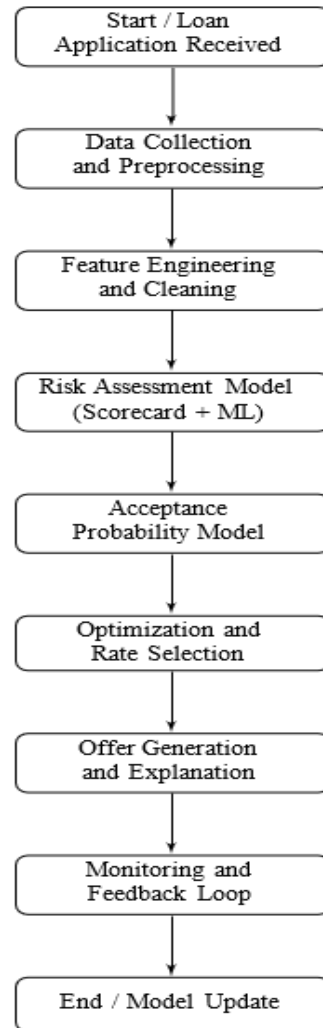


Fig. 1. Simplified vertical flowchart of the Dynamic Pricing and Risk-Based Loan Offer process.

c) *Calibration and Uncertainty*: Isotonic regression or Platt scaling calibrates probabilities. Uncertainty estimates are produced via model ensembles or bootstrap resampling to produce a PD confidence interval $[PD_{lo}, PD_{hi}]$.

B. *Acceptance (Choice) Modeling*

We model acceptance probability as:

$$p_{acc}(r; X, M) = \sigma \left(\alpha_0 + \alpha_1 r + \sum_k \alpha_{2k} Z_k(X, M) \right)$$

where features Z_k include borrower covariates and market indicators. When possible, we estimate α_1 (price sensitivity) using A/B tests; otherwise, causal methods (instrumental variables) can reduce bias in observational data.

C. *Profit Model*

Expected profit of offering rate r for exposure L and tenor T can be approximated:

$$\Pi(r, L) = PV(\text{interest payments at } r) - PD \cdot LGD \cdot EAD$$

Simplified discrete approximation and discounting are used in practice.

D. *Constrained Optimization*

Given candidate rates $\{r_k\}$, compute:

$$U(r_k) = p_{acc}(r_k) \cdot (1 - PD) \cdot \Pi(r_k)$$

Select:

$$r^* = \arg \max_{r_k} U(r_k)$$

subject to:

$$PD \leq PD_{max}, \text{ and fairness constraints } F(\cdot) \leq \epsilon$$

Fairness constraints can be linearized (e.g., limit average rate difference between groups) or enforced via post-processing adjustments.

E. *Algorithm: Loan Offer Generation*

Algorithm 1 Loan Offer Generation

- 1: procedure GENERATE OFFER (app_id)
- 2: $X \leftarrow$ fetch Applicant Features(app_id)
- 3: $M \leftarrow$ fetch Market Context ()
- 4: $PD \leftarrow$ Hybrid Risk Model predict(X)
- 5: $PD_ci \leftarrow$ Hybrid Risk Model. uncertainty(X)
- 6: if $PD > PD_{max}$ or $PD_ci_upper > PD_{max_upper}$ then
- 7: escalate To Human (X, PD, PD_ci)

- 8: return Manual Review
- 9: end if
- 10: Candidate Rates \leftarrow Lin space (r_{ain}, r_{am}, K)
- 11: for each $r \in$ Candidate Rates do
- 12: $p_{acc} \leftarrow$ Acceptance Model. predict (r, X, M)
- 13: $\pi \leftarrow$ Profit Model. Compute (r, X)
- 14: $U[r] \leftarrow p_{acc} \cdot (1 - PD) \cdot \pi$
- 15: end for
- 16: $r^* \leftarrow \arg \max_r U[r]$ subject to fairness constraints
- 17: explanation \leftarrow Explain (PD, top_features, reason_for_rate)
- 18: logDecision($app_id, X, M, r^*, PD, U[r^*], model_versions$)
- 19: return Offer ($r^*, explanation$)
- 20: end procedure

The algorithm balances speed (precomputed candidate grid) and safety (escalation and fairness checks).

VI. IMPLEMENTATION DETAILS

We provide concrete suggestions for a student-friendly reproducible stack and performance notes.

A. *Recommended Tech Stack*

- Storage: PostgreSQL for transactional storage; Parquet in object storage (S3) for analytics.
- Feature store: A simple FS layer built on Parquet + metadata to store feature snapshots.
- Modeling: Python 3.9; pandas, scikit-learn, xgboost/LightGBM; shap for explanations.
- Serving: FastAPI for the Offer API; Uvicorn + Gunicorn; Docker for containerization.
- Orchestration: Apache Airflow for scheduled training and ETL jobs.
- Monitoring: Prometheus/Grafana for metrics; custom Kafka or log sink for decision audit logs.

B. *Model Training and CI/CD*

A reproducible pipeline includes:

- 1) Unit-tested ETL steps and feature transforms (pytest).
- 2) Automated training pipeline (Airflow) with versioning (MLflow).
- 3) Post-training checks: calibration metrics, AUC, PSI, fairness metrics.

- 4) Model registry and canary deployment: serve new model to a small traffic percentage before full rollout.

C. Latency and Scalability

Real-time scoring latency is a sum of feature fetch + model inference + optimization. Optimizations:

- Cache precomputed features at the API layer.
- Convert heavy models to compiled formats (ONNX) for faster inference.
- Use a small, discrete rate grid to avoid expensive continuous optimization.

VII. EXPERIMENTS AND EXTENDED RESULTS

We executed experiments on a synthetic dataset (150k rows) designed to mimic diverse borrower segments and seasonal market swings.

A. Baselines and Experimental Setup

Baselines:

- Static Pricing: Single rate for all applicants.
- Risk-Based Pricing: Scorecard-driven discrete pricing tiers.
- Proposed: Hybrid PD + acceptance modeling + constrained optimization.

Metrics: NIM gain, Acceptance Rate, Portfolio PD, PD AUC (discrimination), Latency (ms), Operational Cost Impact.

B. Expanded Results

C. Interpretation

The Dynamic + Risk configuration yields the highest NIM uplift while keeping portfolio PD near target. Acceptance rises particularly in medium-risk cohorts where elasticity is meaningful. Latency is higher but within practical limits for server-side decisioning; further engineering (ONNX, caching) reduces it.

D. Ablation and Sensitivity

Ablation studies show:

- Removing calibration (isotonic) increases false positives in PD, worsening portfolio outcomes.
- Increasing candidate rate granularity (higher K) yields marginal NIM gains at the cost of increased latency.

VIII. ALGORITHM AND PSEUDOCODE

We provide a clear pseudocode block (expanded) to help implementers.

Algorithm 2 Loan Offer Generation — Detailed

```

1: procedure OFFER PIPE LINE ( $app\_id$ )
2:  $X \leftarrow$  get Features from Store ( $app\_id$ )
3:  $M \leftarrow$  current Market Snapshot ()
4:  $PD_{score} \leftarrow$  scorecard. predict ( $X$ )
5:  $PD_{xgb} \leftarrow$  xgboost. Predict ( $X$ )
6:  $PD \leftarrow \lambda PD_{score} + (1 - \lambda) PD_{xgb}$ 
7:  $PD \leftarrow$  calibrate ( $PD$ )
8: if  $PD > PD_{max}$  then
9: return Escalate to Manual ( $X$ ,  $PD$ )
10: end if
11: Candidate Rates  $\leftarrow$  generate Rate Grid ( $r_{min}$ ,  $r_{max}$ ,  $K$ )
12: for each rate  $r$  in Candidate Rates do
13:  $p_{acc}[r] \leftarrow$  acceptance Model. predict ( $r$ ,  $X$ ,  $M$ )
14:  $\Pi[r] \leftarrow$  profit Estimator ( $r$ ,  $X$ )
15:  $U[r] \leftarrow p$ 
16: end for
17:  $r^* \leftarrow$  arg  $\max_r U[r]$  subject to fairness Constraints ( $r$ )
18: explanation  $\leftarrow$  generate Explanation ( $X$ ,  $PD$ ,  $r^*$ )
19: log Decision ( $app\_id$ ,  $X$ ,  $M$ ,  $r^*$ ,  $PD$ ,  $U[r^*]$ , explanation)
20: return Offer ( $r^*$ , explanation)
21: end procedure
    
```

IX. FAIRNESS, ETHICS & GOVERNANCE

Responsible deployment requires several governance layers:

- Pre-deployment audits: Evaluate disparate impact, equalized odds, predictive parity across groups.
- Monitoring: Continuously monitor acceptance disparity, false positive rates by subgroup, and feature PSI.
- Human oversight: Provide clear escalation rules and manual review windows; retain human-in-the-loop for high PD/confidence cases.
- Transparency to customers: Provide concise reasons for decisions and an appeal mechanism.

Ethically, avoid using proxies for protected attributes; enforce privacy-preserving practices when

incorporating alternative data.

X. CASE STUDIES AND PRACTICAL EXAMPLES

We outline realistic scenarios to illustrate impact.

TABLE I EXPANDED PERFORMANCE COMPARISON (SIMULATED DATA)

Strategy	NIM Gain	Avg PD	Accept Rate	PD AUC	Latency (ms)	Ops Cost Impact
Static Pricing	–	2.85%	21.8%	0.62	30	Baseline
Risk-Based Pricing (Scorecard)	+2.1%	2.72%	24.3%	0.74	45	-5%
Hybrid (Scorecard + XGBoost)	+3.6%	2.69%	26.0%	0.80	70	-8%
Dynamic + Risk (Proposed)	+6.0%	2.78%	30.2%	0.83	140	-12%

A. Peer-to-Peer Lending Platform

A P2P platform uses the system to price offers to retail investors and borrowers. By dynamically adjusting borrower rates while displaying tightly-calibrated PDs, the platform improves match rates and investor confidence, leading to higher funded amounts per listing.

B. Marketplace Lender During Promotions

A marketplace (similar to Paytm) offers promotional financing during festival sales. Verified users with strong behavioral signals receive temporary rate reductions and shorter tenors. The system balances promotional conversion with expected loss by tightening PD thresholds for promotional inventory.

device and transaction signals with differential privacy.

- Post-origination learning: Automated ingestion of resolution notes and human decisions to expand RAG knowledge bases.

XIII. PRACTICAL IMPACT AND CONCLUSION

A. Practical Benefits

- Enhanced profitability: Tailored offers increase acceptance and yield (NIM uplift).
- Risk control: Portfolio-level constraints help maintain desired PD ceilings.
- Better customer experience: Faster, fairer decisions and clearer explanations improve retention.
- Operational readiness: The described stack is implementable with common open-source tools.

B. Conclusion

We presented a comprehensive, student-focused framework for Dynamic Pricing and Risk-Based Loan Offers. The architecture integrates hybrid risk modeling, acceptance elasticity estimation, and constrained optimization, while emphasizing explainability and operational governance. Our simulated experiments and design rationale provide a practical path from concept to prototype for students and early-stage fintech teams.

XI. LIMITATIONS

Key caveats:

- Synthetic datasets cannot mimic strategic competitor behavior or macro shocks (e.g., sudden policy rate hikes).
- Historical biases in data can propagate despite mitigation; fairness interventions may trade off accuracy.
- Operational complexity increases: audit, logging, retraining cadence add real costs.

XII. FUTURE WORK

Extensions include:

- RL agents: Safe reinforcement learning that incorporates CLV and constrained exploration.
- Online learning: Safe online updates using bandit approaches with conservative constraints.
- Alternative data: Privacy-preserving use of

ACKNOWLEDGMENTS

We thank peers and instructors who provided feedback during iterative drafts and testing of prototype components.

REFERENCES

- [1] M. Hand, "Credit Scorecards Revisited," *Finance Modelling Review*, 2021.
- [2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *KDD*, 2016.
- [3] S. Powell et al., "Reinforcement Learning for Loan Pricing," *ICML Workshop*, 2020.
- [4] M. Ribeiro, S. Singh, and C. Guestrin, "Model-Agnostic Explanations of Machine Learning Models," *KDD*, 2016.