# Loan Anomaly Detection Using Unsupervised Ensemble Methods

CS5344 - Big Data Analytics Technology Project

**Team 11:** Roheth Balamurugan(A039399L), Himanshu Maithani (A0314584B)

**Date:** 10th Nov, 2025

# Loan-Level Anomaly Detection Challenge

## Objective

Learn an anomaly scoring function **f: X → [0,1]**

- **f(xi) → 0** for normal loans
- **f(xi) → 1** for abnormal loans
- Trained **only** on normal samples

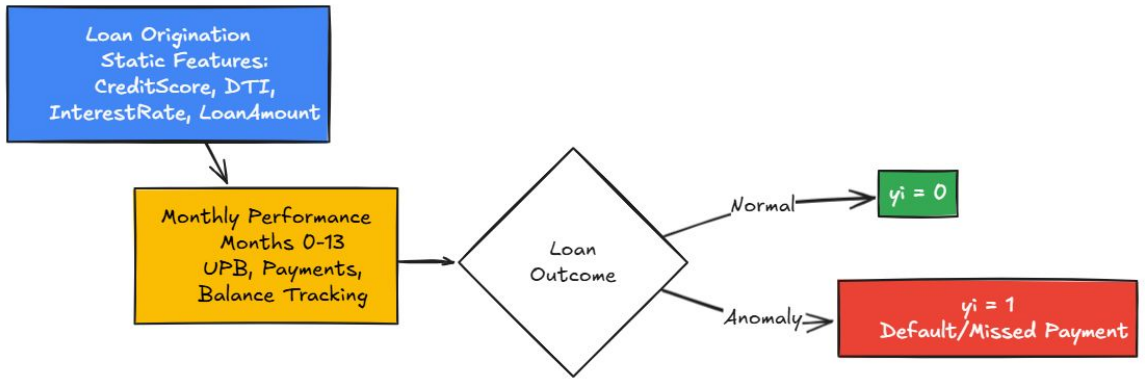
## Business Context

- Detect loans failing to meet monthly payments
- Each loan combines **static borrower info** + **monthly performance**
- Challenge: Identify deviations from typical repayment patterns

## Key Constraints

- Training data contains **only normal loans** (yi = 0 always)
- Use unsupervised anomaly detection methods
- Must generalize to unseen anomaly patterns

## Loan Lifecycle




**Normal**

**Anomalous**

## Evaluation & Challenge

**Primary:** Average Precision (AUPRC)
**Secondary:** AUC-ROC

**"How do we detect anomalies when we've never seen one during training?"**

# Freddie Mac Single-Family Loan Dataset

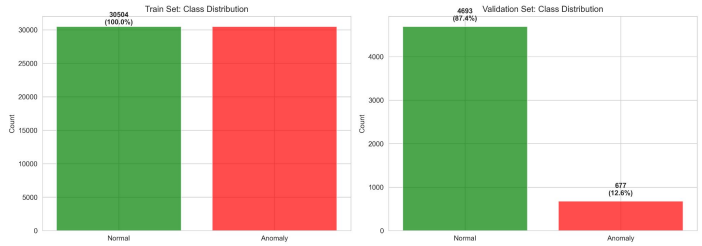
## Data Source

Freddie Mac loan-level dataset (2019)

Data available from 1999-2024 with monthly performance through March 2025

## Data Splits

| Split | Samples (n) | Anomaly Rate | y-labels |
|---|---|---|---|
| **Train** | 30,504 | 0.00% | yi = 0 for all i |
| **Validation** | 5,370 | 12.61%(677 anomalies) | yi ∈ {0,1} |
| **Test** | 13,426 | Unknown | Unlabeled |




## Evaluation Metrics

**Primary:Average Precision (AUPRC)**

**Secondary: ROC-AUC**

## Feature Structure

**m = 145 features** total

### Origination Variables (Static)

- CreditScore
- OriginalDTI
- OriginalInterestRate
- OriginalUPB
- LoanPurpose
- PropertyType
- OccupancyStatus

### Performance Panel (N = 0-13 months)

- N_CurrentActualUPB
- N_InterestBearingUPB
- N_NonInterestBearingUPB
- N_EstimatedLTV
- N_LoanAge
- N_RemainingMonthsToLegalMaturity

## Data Format

Each row i = (si, (ti,1, ri,1), ..., (ti,Ti, ri,Ti), yi)

**si**
Static features

**ti,k**
Time of month k

**ri,k**
Repayment info at month k

**yi ∈ {0,1}**

0 = normal  1 = missed ≥1 payment

# Limitations & Challenges

Key challenges faced during the loan anomaly detection project:

## Data Limitations

- **No anomalies in training** (yi = 0 always) → cannot use supervised learning
- **Limited anomaly examples** in validation (677 out of 5,370 = 12.61%)
- **Potential distribution shift** between validation and test (unknown test anomaly rate)
- **Missing values** in temporal features (especially late months N=10-13)
- **Label definition ambiguity**: "missed ≥1 payment" captures diverse failure modes

## Model Limitations

- **Temporal dependencies underexploited**:
  - Treated as independent time windows, not sequential trajectories
  - No explicit loan lifecycle modeling (LSTM, RNN, attention)
- **Linear dimensionality reduction** (PCA) → may miss nonlinear manifolds

## Explainability

- **Threshold selection** and hyperparameter tuning fusing validation set.
- **Explainability** for individual predictions using Isolation forest and shap values

# Evaluation Metric: Why Average Precision (AUPRC)?

## The Problem: AUC-ROC is Misleading with Imbalanced Data

AUC-ROC rewards models for correctly identifying **True Negatives** (the normal loans). With 87% of our data being normal, a model can get a high AUC-ROC score just by being good at spotting the easy majority class, even if it fails to find the rare anomalies.

## The Solution: AUPRC Focuses on What Matters

AUPRC is built on **Precision** and **Recall**, which are ideal for this task:

- **Precision**: "Of all the loans we flagged, how many were actually anomalies?"
  Measures the **quality** of our flags. High precision means fewer false alarms.
- **Recall**: "Of all the true anomalies that exist, how many did we find?"
  Measures the **completeness** of our detection. High recall means we miss fewer bad loans.

## Why AUPRC is the right tool:

It focuses **exclusively** on the performance of the positive (anomalous) class. It heavily penalizes **False Positives**, making it a much more realistic measure of success.

# Exploratory Data Analysis: Missing Values

## Challenge & Sentinel Values

**Systematic missing value / outside range patterns** detected in raw data

**Sentinel Values (encoded missingness / outside range):**

- CreditScore: **9999** → NaN
- OriginalDTI: **999** → NaN
- OriginalLTV: **999** → NaN
- MI_Pct: **999** → NaN
- Flag fields: **'9'**, **'99'** → NaN

## Key Insight

### Missingness itself can be informative

A missing credit score might indicate a higher-risk borrower profile, preserving valuable information about borrower characteristics.

## Handling Strategy

### Missing Indicator Features

Create binary flags to explicitly signal missing values

### Median Imputation

For numeric features, impute using median from training data (it is a robust statistic that is not easily skewed by outliers, unlike the mean)

### Mode Imputation

For categorical features, impute using mode from training data

## Missing Values Pattern

# Data Preprocessing Strategy

## Step-by-Step Pipeline


### Sentinel Mapping

- Replace **9999** (CreditScore), **999** (DTI, LTV, MI_Pct) with **NaN**
- Create **missing indicator flags** to preserve signal of missingness
- Preserves potentially informative patterns of missing values

### Categorical Encoding

- Applied to **LoanPurpose, PropertyType, OccupancyStatus**
- Uses **LabelEncoder** with special handling for unseen categories
- Creates **UNKNOWN** category for test set values not seen in training

### Imputation

- **Numeric:Median imputation** (robust to outliers)
- **Categorical:Mode imputation** (most frequent category)
- Fitted **only on training data** to prevent data leakage

### Scaling

- Applies **StandardScaler** to feature matrix
- Transforms data to **zero mean and unit variance**
- Essential for distance-based anomaly detectors

# Temporal Feature Engineering

## Challenge

Raw performance panel has **14 monthly snapshots** (N=0-13) but we need to extract meaningful trajectory patterns to identify payment stress.

## Multi-Window Strategy

**Main: Quarterly Sampling**
Months **[0, 3, 6, 9, 12]**

**Alt1: Bimonthly Sampling**
Months **[0, 2, 4, 6, 8, 10, 12]**

**Alt2: First-Year Focus**
Months **[0, 3, 6, 9]**

## Results

**~60-80** additional temporal features capturing loan lifecycle dynamics

## Features per Window

### Trend
(last - first) / |first|
Captures overall direction

### Volatility
std / |mean|
Indicates stability

### First-diff Mean
Avg(Δt)
Average change

### First-diff Std
Std(Δt)
Change variability

**Applied to Variables:**

UPB | InterestBearingUPB | LTV | LoanAge

## Intuition

Increasing **volatility** or **negative trends** in later months signal payment stress.

# Domain-Specific Feature Engineering

## 1. Amortization Shortfall Signals

**Motivation:** Normal loans follow predictable principal reduction schedules. **Deviations signal issues.**

**Calculation:**

**Expected principal reduction:** Annuity formula with InterestRate and RemainingMonths

**Observed principal reduction:** Month-to-month change in UPB(Δ(UPB))

**Shortfall ratio = (Expected - Observed) / Expected**

**Engineered Features:**

- **amort_short_mean:** Average shortfall across all periods
- **amort_short_70:** Fraction of periods with >70% shortfall (severe)
- **amort_short_50:** Fraction of periods with >50% shortfall (moderate)

## 2. PCA Dimensionality Reduction

**Challenge:** Increased dimensionality from feature engineering can lead to:

- Higher computational complexity
- Increased noise in distance-based algorithms
- Curse of dimensionality effects

**Implementation:**

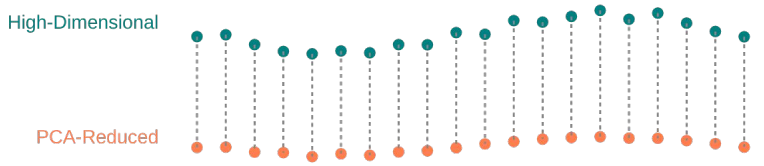
Reduce **~166 features** to **80 principal components**

Retains **~95%** of the original variance

**Benefits:**

**Noise reduction:** Particularly beneficial for distance-based algorithms like LOF and k-distance

**Dimensionality mitigation:** Improves efficiency and effectiveness of detectors



# Domain-Specific Feature Engineering

## Amortization Features Creation

We estimate what a normal loan's monthly principal payment should be, based on its balance, interest rate, and remaining term.

## The Annuity Formula

The **Expected Monthly Payment** is the foundation of our calculation, determined by the loan's current state:

Formula:

$$\text{Expected Payment} = P \times \frac{r(1+r)^n}{(1+r)^n - 1}$$

Where:

- P = Current loan balance (from **N_InterestBearingUPB**)
- r = Monthly interest rate (from **N_CurrentInterestRate**)
- n = Remaining months (from **N_RemainingMonthsToLegalMaturity**)

## Key Points

- **Observed vs. Expected:** We compare actual principal reduction against this expected value to spot payment shortfalls.
- **Applicability Mask:** Logic applies only to standard loans excludes Interest-Only or special products to prevent false alarms.
- **Purpose:** Detects repayment stress early by flagging loans that deviate from their expected amortization path.

# Baseline Model Evaluation

## Objective

Systematically evaluate unsupervised anomaly detection algorithms to identify best approaches for loan anomaly detection.

## Algorithms Tested (7 families)

### Local Outlier Factor
Measures local density deviation relative to neighbors

### k-distance
Distance to k-th nearest neighbor

### Isolation Forest
Tree-based isolation method

### Elliptic Envelope
Robust covariance outliers

### One-Class SVM
Maximum margin separation

### DBSCAN
Density-based clustering

### PCA Reconstruction Error
Linear subspace deviation

LOF consistently outperforms other methods across various configurations

## Configurations

### Preprocessing Setups
- robust_pca80
- standard_pca80
- robust_pcaNone

### Hyperparameter Tuning
- LOF: k=5,7,10
- k-distance: k=3,5,7
- Multiple configurations per algorithm

### Total Configurations
**~50 baseline configurations tested**

## Evaluation Metrics

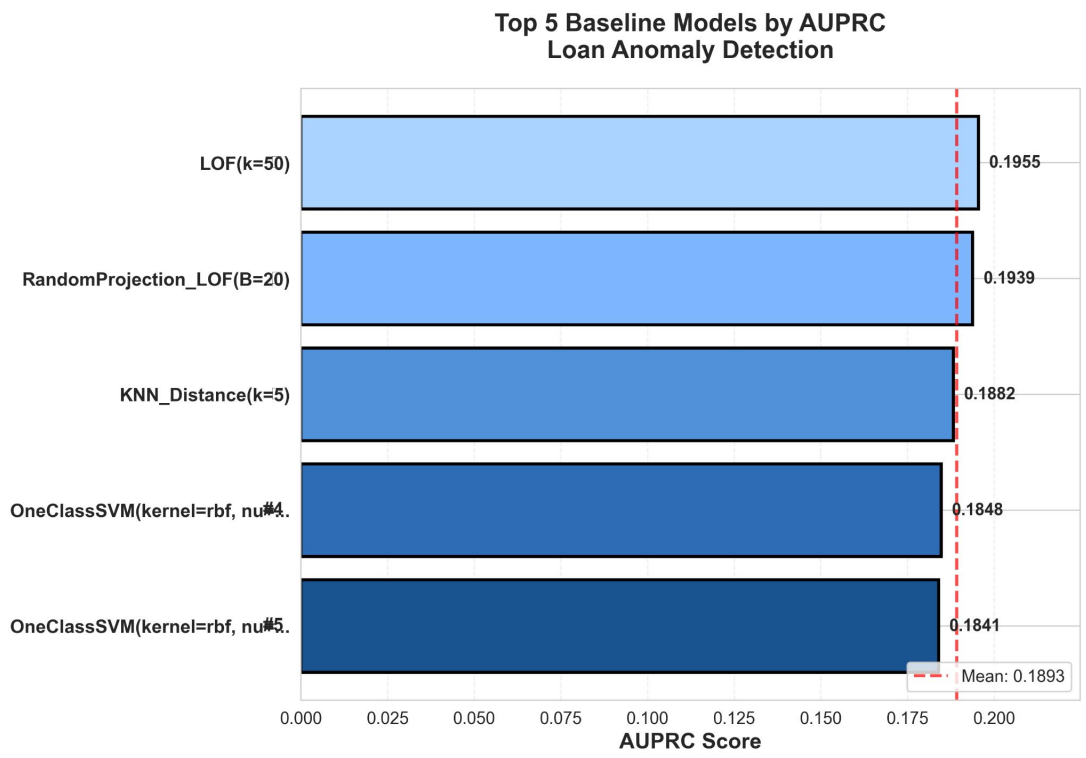
**AUPRC**
Primary

**AUROC**
Secondary

**F1@best-threshold**
Reference

# Baseline Model Performance

## Top Models by AUPRC




## Performance Metrics

| Rank | Model | AUPRC | AUROC |
|---|---|---|---|
| 1 | LOF(k=5) | 0.195483 | 0.564805 |
| 2 | RandomProjection_ LOF(B=20) | 0.193859 | 0.562951 |
| 3 | KNN_Distance(k=5) | 0.188237 | 0.554002 |
| 4 | OneClassSVM(kern el=rbf, nu=0.1) | 0.184753 | 0.555603 |
| 5 | OneClassSVM(kern el=rbf, nu=0.1) | 0.18405 | 0.537535 |

Local Outlier Factor (LOF) with k=50 neighbors achieved the best baseline performance (AUPRC: 0.1955), demonstrating that distance-based neighborhood methods outperform isolation and boundary-based approaches for this loan anomaly detection task.

# Local Outlier Factor: Algorithm Intuition

## How LOF Works

Measures **local density deviation** relative to neighbors:

**LOF = (Average Neighbor Density) ÷ (Point Density)**

**LOF ≈ 1:**
Normal point (similar density to neighbors)

**LOF > 1:**
Anomaly (lower density than neighbors)

## Why Effective for Loan Anomalies

**Captures Multi-Dimensional Patterns**

Identifies loans with unusual **feature combinations**, not just extreme individual values

Low Score | High DTI | High Rate

→ **Together: Anomalous (borderline individually)**

**k-Parameter**

Controls neighborhood size for density calculations

**k=5-10** is optimal for this dataset

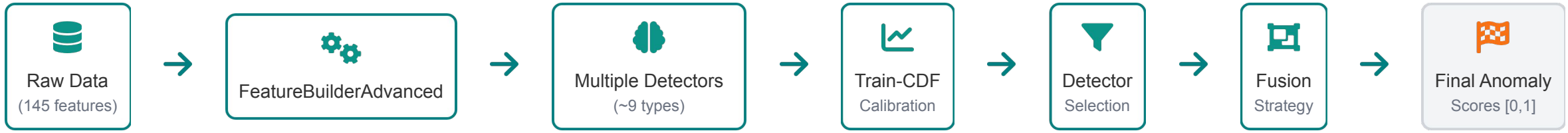
Normal Density

Anomaly (Lower Density)

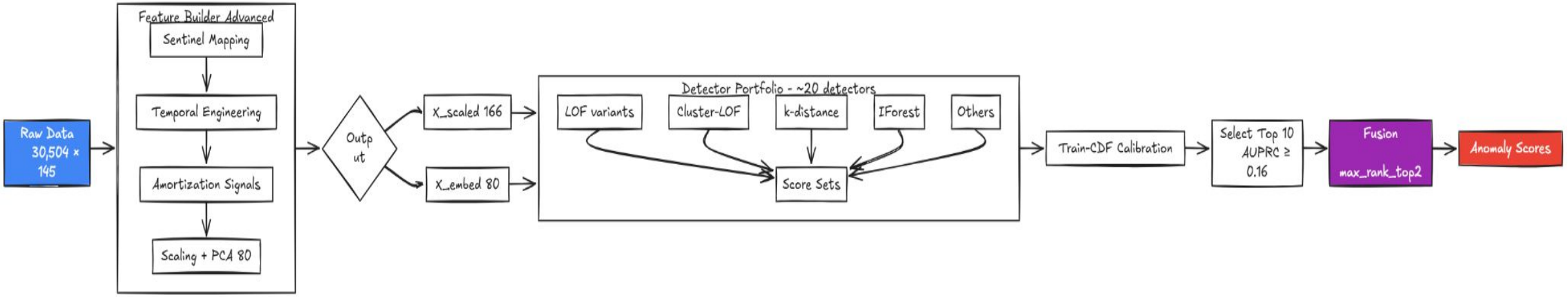
Too small (k=3): noisy results; Too large (k=20): misses local patterns

# Final Model: Ultra Unsupervised Ensemble

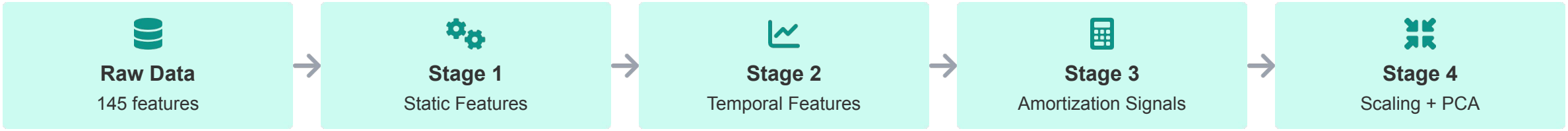
## High-Level Architecture




Implementation: final_approach/unsup_ultra_ensemble_fast_improvement.py



# FeatureBuilderAdvanced Class




## Stage 1: Static Features

- **Sentinel Mapping:** Replace with NaN + missing indicators
- **Categorical Encoding:** LabelEncoder with UNKNOWN handling
- **Imputation:** Median for numeric, Mode for categorical

## Stage 2: Temporal Features

- **Multi-window extraction:** Main, Alt1, Alt2 strategies
- **Features:** Trend, Volatility, First-diff mean/std
- **Target:** UPB, InterestBearingUPB, LTV, LoanAge
- **Result:** ~60-80 temporal features

## Stage 3: Amortization Signals

- **Expected vs Observed:** Annuity formula calculation
- **Features:** amort_short_mean, amort_short_70, amort_short_50
- **Masking:** Applied only to FRM loans (excludes interest-only, balloon)

## Stage 4: Scaling + PCA

- **StandardScaler:** Zero mean, unit variance
- **X_scaled:** ~166 features (full feature set)
- **PCA:** 80 components, ~95% variance retained
- **X_embed:** Reduced feature space for modeling

# Ensemble Detector Components

The ensemble leverages **9 detector families** with to capture different types of anomalies in loan data.

### LOF

**Purpose:** Local density deviation

**Variants:** k ∈ {4,5,6,7,8,10,12}

**Input:** X_embed (PCA)

### Cluster-LOF

**Purpose:** Cohort-specific anomalies

**Variants:** n_clusters=12

**Input:** X_embed (PCA)

### k-distance

**Purpose:** Distance to k-th neighbor

**Variants:** k ∈ {3,5,7,9,11}

**Input:** X_embed (PCA)

### Isolation Forest

**Purpose:** Tree-based isolation

**Variants:** n_trees=500

**Input:** X_embed (PCA)

### Elliptic Envelope

**Purpose:** Robust covariance outliers

**Variants:** contamination=0.1

**Input:** X_embed (PCA)

### PCA Reconstruction

**Purpose:** Linear subspace reconstruction error

**Variants:** 80 components

**Input:** X_scaled (full)

### Random Projection LOF

**Purpose:** Bagged LOF in random subspaces

**Variants:** n_bags=40

**Input:** X_scaled (full)

### Mahalanobis

**Purpose:** Global covariance-based distance

**Variants:** Standard covariance

**Input:** X_scaled (full)

### Amortization Signal

**Purpose:** Payment shortfall aggregation

**Variants:** Custom

**Input:** X_scaled (amort slice)

Detectors with validation AUPRC ≥ 0.16 (typically top 10)

# Training on Normal Loans Only

## Fundamental Principle

All detectors fit **ONLY** on Dtrain:

- Training set contains **30,504 normal loans** (yi = 0 always)
- No labeled anomalies during training phase
- Adheres to **unsupervised learning** paradigm

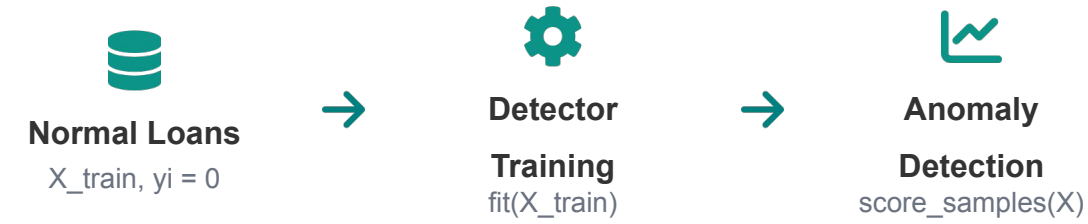
## Why This Works

- Unsupervised detectors learn the **distribution of normal loans**
- At inference: **deviations from normal** → high anomaly scores
- No labeled anomalies needed during training phase
- Example: LOF (Local Outlier Factor)

## No Data Leakage

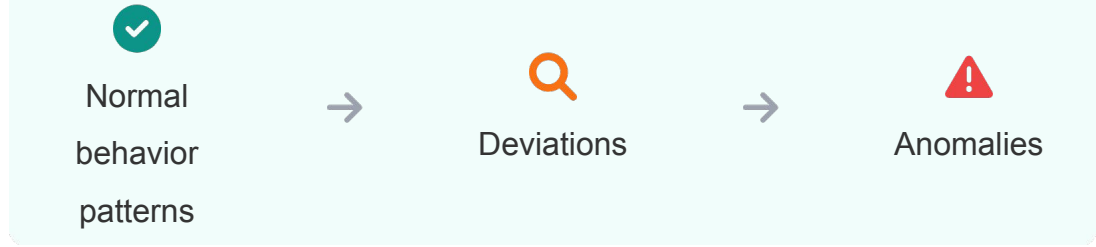
- Validation/test data **NEVER** used for fitting
- Used **ONLY** for scoring and hyperparameter selection
- Strict adherence to unsupervised learning constraints

## Training Process




## Key Insight

Training on normal loans = learning **"typical loan behavior"**



# Train-CDF Calibration

## The Problem

Different detectors produce incomparable score scales:

- **LOF**: ~1-5 (local density ratio)
- **k-distance**: ~0.1-10 (Euclidean distance)
- **Isolation Forest**: ~-0.5-0.5 (tree depth)

**Cannot** directly combine or compare these scores

## The Solution: Train-CDF Calibration

- Fit **empirical CDF** on **training set scores only**
- Map any score x → **percentile** in training distribution → [0,1] probability
- Higher percentile = more extreme relative to normal = higher anomaly probability

## Advantages

- No validation leakage (uses **ONLY** training distribution)
- Normalizes all detectors to [0,1] scale
- Preserves relative ordering of anomalies

## Implementation:

In practice, we compute the empirical CDF by first sorting all detector scores from the training set. For any new score x, we find its position (rank) within this sorted list and divide it by the total number of training scores. This converts every detector's raw output into a percentile-based probability between 0 and 1 allowing consistent comparison and fusion across all detectors while strictly using only training data (no leakage).

$$F_{\text{train}}(x) = \frac{\text{Number of training scores } \leq x}{\text{Total number of training scores}}$$

# Ensemble Fusion Approaches

Combining detector outputs using three strategies:

## Rank-Based Fusion

Normalizes scores to [0,1] range

- **Weighted Average:** ∑(wi × ranki)
- **Max Rank:** max(rank1, rank2, ..., rankD)
- **Max Rank Top-k:** Uses top-2 or top-3 detectors

## Probability-Based Fusion

Uses Train-CDF calibrated probabilities (pi ∈ [0,1])

- **Weighted Average:** ∑(wi × pi)
- **Noisy-OR:** 1 - ∏(1 - pi)
- **Max Probability:** max(p1, p2, ..., pD)

## Cohort-Normalized Fusion

Addresses cluster-specific scales

- **Clustering:** KMeans with n_clusters=12
- **Normalization:** Z-score within cohorts
- **Fusion:** Apply on normalized scores

## Selection Process

All proposed fusion strategies are evaluated on the validation dataset (Dvalid). The strategy that yields the best AUPRC is selected for the final model.

# Hyperparameter Optimization

Key hyperparameters selected through systematic evaluation:

## PCA Components

Dimensionality reduction components

Tested: 60, 80, 100, 120

**Selected:** 80 (best validation AUPRC)

## LOF k-values

Local Outlier Factor neighborhood sizes

Tested: k=4,5,6,7,8,10,12,15

**Selected:** k=4-12 (multiple variants)

## Detector Selection Threshold

Filtering criteria for detector selection

AUPRC ≥ 0.16

**Result:** ~10 detectors selected

## Clustering

For cohort normalization

Tested: 10, 12, 15 clusters

**Selected:** 12 clusters

## Validation Strategy

Used validation AUPRC to select hyperparameters, keeping the selection process simple and threshold-based to avoid overfitting.

# End-to-End Pipeline Architecture

### Input

Raw loan data(30,504 × 145)

### FeatureBuilderAdvanced

- Sentinel mapping → NaN + flags
- Temporal engineering (3 window strategies)
- Amortization signals (shortfall features)
- Scaling + PCA (80 components)

### Detector Training

on Dtrain only (yi = 0)

- 7× LOF (k=4,5,6,7,8,10,12)
- 5× k-distance
- 1× PCA reconstruction
- 1× Cluster-LOF
- 1× Isolation Forest
- 1× Amortization signal

### Score Generation

Apply detectors to:

**Dvalid** **Dtest**

### Train-CDF Calibration

Scores → [0,1] probabilities

### Detector Selection

Keep top 10 detectors

AUPRC ≥ 0.16

### Fusion & Output

**Evaluate fusion strategies**

**Output:** Final anomaly scores [0,1]

# Initial Tried Configuration

## Hyperparameters

- **PCA Components:** 80
  Retains ~95% variance
- **Detector Selection Threshold:** AUPRC ≥ 0.16
  Results in ~10 detectors
- **Number of Detectors Selected:** 10
  Top performers by validation AUPRC

## Top 10 Detectors

| Rank | Detector | AUPRC | Notes |
|---|---|---|---|
| **1** | **LOF k=5** | **0.2988** | Best overall |
| **2** | **LOF k=7** | **0.2965** | Close second |
| 3 | LOF k=10 | 0.2943 | Robust |
| 4 | Cluster-LOF | 0.2881 | Cohort-specific |
| 5 | k-distance k=3 | 0.1949 | Complementary |
| 6-8 | k-distance k=5,7,9 | 0.1929-0.1890 | Distance-based |
| 9 | Isolation Forest | 0.1817 | Tree-based diversity |
| 10 | PCA reconstruction | 0.1742 | Linear subspace |

# Final Selected Configuration

## Hyperparameters

- **LOF: n_neighbors** 10–50, feature bagging 60–80%, 5–10 random projections
- **RFOD: n_estimators** 25-160, temporal windows 3–5, top-q 10–15%
- **Fusion: train-CDF calibration + EVT shaping;** amort-gated fusion emphasizes strong amortization signals

| Rank | Detector | AUPRC | Notes |
|---|---|---|---|
| 1 | **amort** | 0.4748 | Amortization shortfall signal; strongest single detector. |
| 2 | **bag_lof_embed** | 0.3419 | Feature-bagged LOF on PCA embedding; best LOF variant. |
| 3 | **cohort_lof_embed** | 0.3191 | Cluster-specific LOF (per KMeans cohort); captures local deviations. |
| 4 | rp_lof_embed | 0.3054 | Random-projection LOF bagging; adds robustness via subspace averaging. |
| 5 | rfod_temporal | ~0.27 | Temporal RFOD; modest gains, complementary to amortization. |
| 6 | rfod (global) | ~0.25 | Global RFOD on high-variance features; struggled with correlated inputs. |
| 7 | iforest_embed | ~0.23 | Isolation Forest on PCA embedding; generic density baseline. |
| 8 | ocsvm_embed | ~0.22 | One-Class SVM on embedding; adds marginal diversity. |
| 9 | kdist_multi_mean | ~0.20 | Multi-k distance (kNN) metric; baseline distance-based detector. |

# Intermediate Results

## Performance Comparison

| Approach | Best AUPRC | AUROC | Notes |
|---|---|---|---|
| Baseline LOF (single, k=5) | 0.2988 | 0.6607 | Simple baseline |
| Cluster-wise LOF | 0.3324 | 0.7629 | Per loan-type clusters |
| **Final Ultra Ensemble** | **0.4524** | **0.7597** | |

## Why Final Ultra Ensemble is Superior

**Diversity**
Captures different anomaly patterns

**Robustness**
Reduces variance compared to single detector

**Calibration**
Prevents score scale issues

**Systematic Evaluation**
All fusion strategies tested

**Trade-off**

Slight AUPRC change for improved robustness to distribution shift

# Results

## Performance Comparison

| Phase | Approach | Best AUPRC | Key Components |
|---|---|---|---|
| Phase 1 | Baseline LOF (k-tuned) | 0.19-0.20 | Standard preprocessing + PCA (80d) + single detector |
| Phase 2 | FeatureBuilderAdvanced + Multi-Detector | 0.25-0.35 | Temporal features, amortization shortfall, train-CDF calibration |
| | Amortization Detector (single) | 0.41 | Domain-specific shortfall signal |
| Phase 3 | **Ultra Ensemble** | **0.45 in kaggle** | 11 detectors: amort-gated weighted fusion |
| | **RFOD Hybrid (Final)** | **0.47 in kaggle** | RFOD-temporal + global + ensemble integration |

## Why Ultra Ensemble Performed Well

- Signal complementarity across density, distribution, and domain-specific methods
- Amortization gating preserved strongest signal (0.4748) while adding complementary detectors
- Train-CDF calibration enabled effective fusion

## Why RFOD Hybrid Was Superior

- Predictive residuals captured feature interdependency violations missed by density methods
- Uncertainty quantification (residual + tree variance) identified rule violations and uncovered regions
- Temporal RFOD detected payment evolution patterns critical for emerging defaults

# Future Enhancements

Exploring advanced modeling approaches to enhance loan anomaly detection

## Enhance Feature Richness

- Create interaction features (e.g., CreditScore × UPB trend) to detect unexpected borrower behavior.
- Add stability metrics showing how consistently a loan stays near its average UPB or rate.

## Diversify the Detector Ensemble

- Refine cohort normalization with more granular KMeans clustering.
- Test additional detectors like Elliptic Envelope and k-distance variants for broader anomaly coverage.

## Optimize Ensemble Combination

- Compare fusion rules (e.g., noisy-OR, max-rank-top3, weighted-average-top2) using validation AUPRC.
- Apply probability-based fusion by mapping detector scores via training-set CDFs.
- Select the fusion strategy that gives the best overall AUPRC and model stability.

# Experiments

We went through a few phases.

## Phase 1 – Baselines

- Simple pipeline:
  - Preprocessing - PCA (80D)
  - single detectors (LOF, kNN-distance, Isolation Forest, One-Class SVM).
- Result: Best LOF model achieved AUPRC ~0.19–0.20, slightly above random.

## Phase 2 – FeatureBuilderAdvanced & Ensemble

- Introduced richer features: missing flags, temporal trends/volatility, first-diff, amortization shortfall.
- Applied robust scaling + PCA, then combined multiple detectors (LOF, Cluster-LOF, kNN, IF, PCA-Recon).
- Used train-CDF calibration and rank/probability-based fusion.
- Result: Ensemble improved to AUPRC 0.40–0.45.
- Key insight: amortization shortfall and cluster-aware LOF were consistently top signals.

## Phase 3 – RFOD Hybrids & Stronger Detectors

- Added RFOD (RandomForest-based Outlier Detection) for residual-based anomaly scoring (global & temporal).
- Introduced CAD (Correlation anomaly detection) for abnormal feature co-movements and RP-LOF for robust density via subspace bagging.
- Refined clustering (KMeans on PCA embedding), cohort-level LOF, OCSVM, IF, and HBOS.
- Amortization-gated fusion yielded best results.
- Result: Amortization detector alone reached ~0.47–0.48, final ensemble peaked at ~0.49–0.50 AUPRC.
- Tried robust PCA and granular clustering marginal gains, dropped for simplicity.

# Explainability: What Drives Anomaly Detection?

**Method: Unsupervised explainability using IsolationForest + SHAP + permutation importance on unlabeled train data to identify which features drive "anomalousness."**

### 1. Static Categorical Features (Top Signal)

- LoanPurpose – #1 in both SHAP & permutation; refinance patterns correlate with borrower stress
- PropertyType, SellerName, Channel, ProgramIndicator – origination characteristics reflect portfolio risk

### 2. Temporal Loan-Behavior Dynamics

- EstimatedLTV volatility/trends (w_main_vol, w_alt2_vol, w_alt1_trend) – LTV fluctuations signal unstable property value or payment issues
- InterestBearingUPB_w_main_dmean, RemainingMonthsToLegalMaturity – deviations from normal amortization schedule indicate repayment problems

### 3. Amortization Shortfall (Our Domain Feature)

- amort_short_mean, amort_short_50, amort_short_70 – ranks 14, 19, 24 (SHAP); 16, 19, 22 (permutation)
- Validates our domain signal: models independently identify these as high-impact without labels

### 4. Origination Risk Metrics

- OriginalLTV, OriginalCLTV, OriginalDTI, CreditScore – classical borrower risk, interact with temporal volatility

### 5. Missingness as Signal

- ProgramIndicator_missing – high importance; missing metadata itself indicates abnormal loans

**Takeaway: Temporal volatility + amortization patterns + origination context drive unsupervised anomaly detection, validating our feature engineering.**

# Lessons Learned

Key insights that shaped our final solution and guided what truly mattered beyond model complexity.

## Domain Knowledge > Raw Complexity

- The key breakthrough was the amortization shortfall feature — derived from loan repayment mechanics.
- Outperformed most generic unsupervised models.

> **Strong domain features beat complex base models in structured finance.**

## Ensembles Need Calibration, Not Just Stacking

- Raw score averaging failed due to inconsistent ranges/tails.
- **Train-CDF calibration + EVT shaping** aligned scales, emphasized extremes.
- **Amort-gated fusion** let the amortization detector dominate when signals were strong.

## Why RFOD still has room for improvement

- RFOD (Random Forest–based Outlier Detection) showed good results but with scope for improvement:
  - Feature subset constraint: Used only 25-30 of 140+ features due to computational cost
  - Complex correlations: Engineered amortization/volatility features created dependencies that limited residual learning
  - Training overhead: Multiple feature-specific forests added significant time vs. density methods

> **RFOD achieved ~0.50 AUPRC but feature selection trade-offs likely capped potential; full-feature training could improve results.**

## Not all extra complexity pays off

- Tried feature bagging, robust PCA, more detectors - minimal gains.
- Real improvements came from:
  - Quality domain features
  - Compact, diverse ensemble
  - Well-designed fusion

> **Focus on smarter calibration and features, not model count.**

# Thank you