AI Data Scientist Case Study

Minjae Seo

2026-01-20

Q1. Lasso (L1) Regularization and Multicollinearity

Lasso regression performs variable selection through its L1 penalty. The objective function is $\mathcal{L}(\beta) = \sum_{i=1}^{n}[-y_i \log \hat{p}_i - (1-y_i)\log(1-\hat{p}_i)] + \lambda \sum_{j=1}^{p} |\beta_j|$. The key mechanism is the geometry of the constraint region. The L1 penalty creates a **diamond shaped** constraint in coefficient space, and when the loss contours intersect this region they often hit a **vertex**, forcing some coefficients to zero. L2 (Ridge) creates a circular constraint, so coefficients shrink toward zero but rarely reach it exactly. In my experiment with 100 features (10 informative), C=0.01 (lambda=100) kept 6 features while C=0.1 kept 15, showing how stronger regularization increases sparsity.

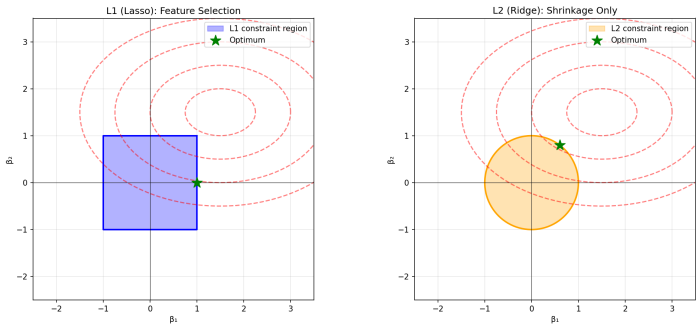

Figure 1: L1 vs L2 Geometry

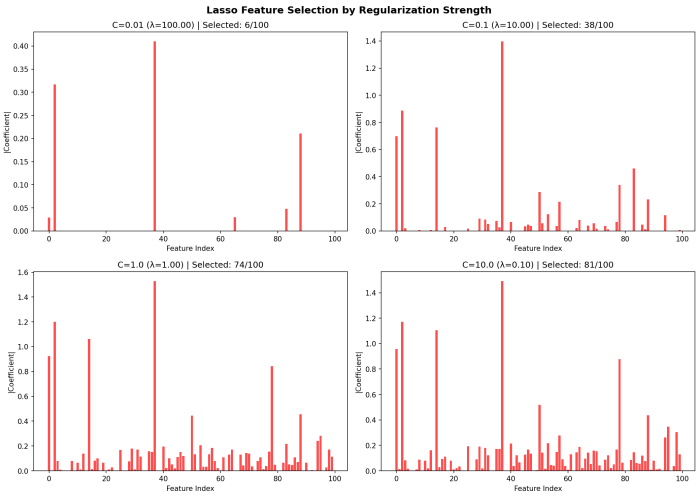

Figure 2: Lasso Feature Selection by

Regarding multicollinearity, linear and tree based models respond differently. I ran 50 bootstrap iterations using synthetic credit data with high correlation between annual income and monthly income (r=0.92) and between annual income and credit limit (r=0.91). For logistic regression, multicollinearity flattens the likelihood surface, so many coefficient combinations yield similar predictions. This leads to coefficient instability: annual income showed CV 135% and credit limit CV 239% across samples, making individual variable interpretation unreliable. Random forest is more robust because trees split sequentially on features; correlated variables create similar partitions without destabilizing predictions (CV about 5 to 7%). The tradeoff is that importance scores spread across correlated features, making each appear less critical. I recommend L1 regularized logistic regression when interpretability is required and tree based models when predictive performance is the priority.

Summary: Logistic regression is highly sensitive to multicollinearity and has unstable coefficients, so interpretation is difficult. Random forest shows low sensitivity and more stable importance, so interpretation is moderate.

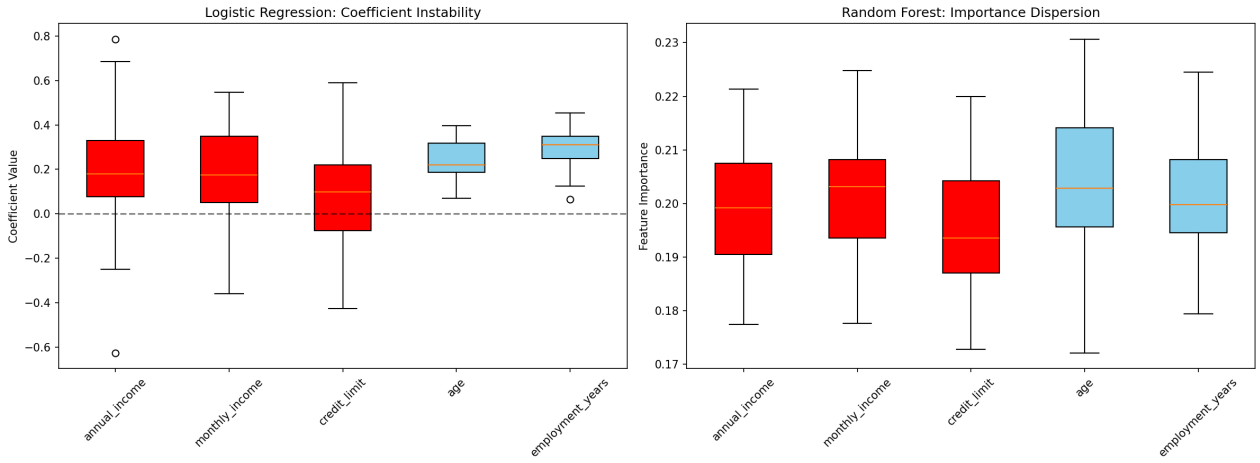

Figure 3: Coefficient/Importance Stability Comparison

Q2. Class Imbalance: Metrics and Handling Methods

In imbalanced classification problems like credit default prediction, accuracy is misleading. With a 5.4% default rate in my synthetic data, a naive classifier that predicts all customers as non defaulters achieves 94.6% accuracy but catches zero actual defaulters (recall 0%, F1 0%). This accuracy paradox happens because accuracy is dominated by the majority class and ignores the minority class we care about detecting.

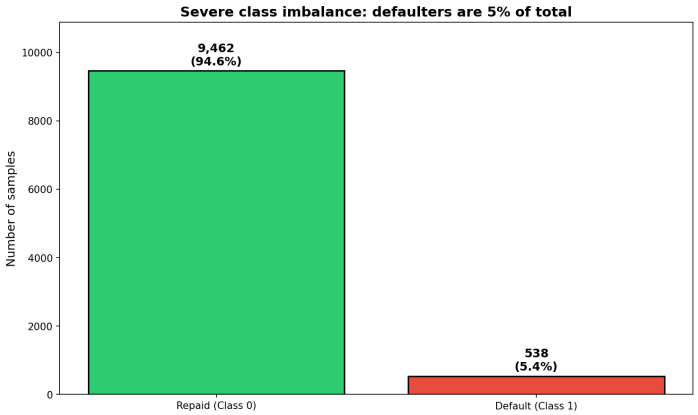

Figure 4: Class Distribution (5.4% default)

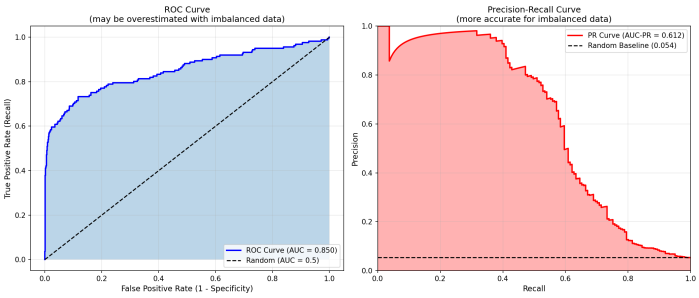

Figure 5: ROC vs PR Curve Comparison

F1 score and PR AUC are better metrics for this problem. F1 score, defined as $\frac{2 \times Precision \times Recall}{Precision + Recall}$, is the harmonic mean of precision and recall and penalizes models that sacrifice one for the other. PR AUC is valuable because its baseline equals the class prevalence (5.4%), which reflects minority detection capability. In my experiment, ROC AUC was 0.850, which looks strong, but PR AUC was 0.612 and revealed the true difficulty.

When comparing imbalance handling methods, I found cost sensitive learning more robust than SMOTE. SMOTE generates synthetic minority samples by interpolating between existing examples, which can create economically implausible borrowers in sparse regions and amplify noise near class boundaries. Cost sensitive learning instead modifies the loss function to penalize minority errors more heavily, preserving the original data distribution. My cross validation experiment (5 fold x 10 repeats) showed that while both methods achieved similar F1 scores (SMOTE 0.335, cost sensitive 0.323), cost sensitive learning had lower variance (sigma 0.017 vs 0.020). This stability matters in production where models face distribution drift; those trained on real data generalize better than those fitted to synthetic samples.

Summary: SMOTE had F1 mean 0.335 with std 0.020 (CV 5.8%). Cost sensitive learning had F1 mean 0.323 with std 0.017 (CV 5.3%).

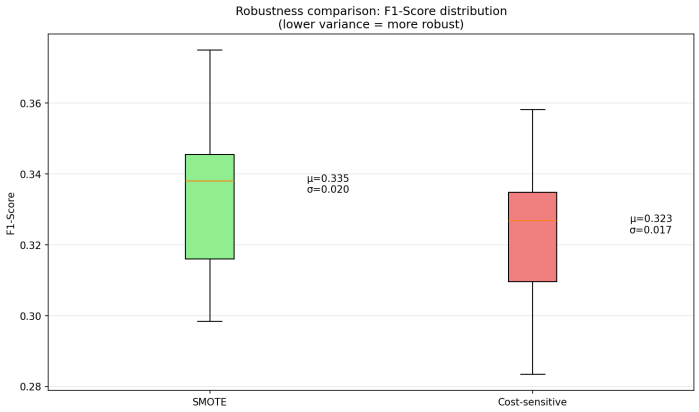

Figure 6: F1-Score Variance Comparison

Q3. Bias Variance Tradeoff and Concept Drift Detection

When a model performs well on training data but degrades in production, the bias variance tradeoff provides a diagnostic framework. In my experiment, an overfitted random forest (no depth limit) achieved 100% training accuracy but only 92.7% test accuracy (7.3% gap), while a properly regularized model showed 91.4% training and 91.3% test accuracy (0.1% gap). The learning curve clearly distinguishes these cases: a persistent gap between training and validation scores indicates high variance, where the model memorizes training patterns rather than learning generalizable relationships. However, for credit risk models in production, the primary cause of performance degradation is often **Concept Drift** rather than simple overfitting, because customer behavior and macroeconomic conditions evolve over time.

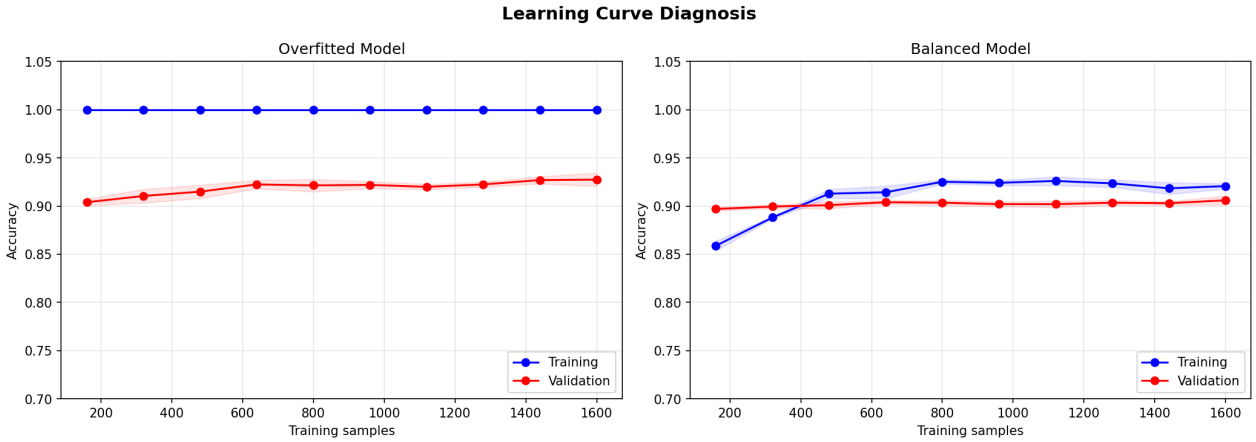

Figure 7: Learning Curve: Overfitted vs Balanced Model

To detect Concept Drift, I propose using PSI (Population Stability Index) and CSI (Characteristic Stability Index). PSI measures input distribution shift using the formula $PSI = \sum_i (q_i - p_i) \ln(q_i/p_i)$, where p_i and q_i are the proportions in reference and current distributions respectively. The interpretation thresholds are: PSI $<$ 0.1 (stable), 0.1 to 0.25 (monitor closely), and $>=$ 0.25 (action required). CSI applies the same calculation per feature to identify which variables are driving the drift. In my simulation modeling a recession scenario (2020 Q1 through 2021 Q2), PSI remained stable through Q2 (0.077), breached the threshold in Q3 (0.254), and escalated sharply during the recession (Q4: 1.020, 2021: 1.32 to 1.97). CSI analysis pinpointed annual income (CSI 1.97) as the primary drift driver, while age and debt ratio remained stable (CSI $<$ 0.02), which provides actionable insight for model updates.

Summary PSI timeline: 2020 Q1 0.000 (stable), 2020 Q2 0.077 (stable), 2020 Q3 0.254 (retrain), 2020 Q4 1.020 (urgent), 2021 Q1 to Q2 1.32 to 1.97 (urgent).

For retraining frequency, I recommend a hybrid approach that combines scheduled and trigger based retraining. Scheduled retraining should occur quarterly using a rolling 12 to 24 month data window to capture gradual drift while maintaining sufficient training volume. Trigger based retraining activates when PSI $>=$ 0.25 on key features (review within 1 to 2 weeks) or when F1 or PR AUC drops more than 10% (emergency retraining within days). The monitoring framework

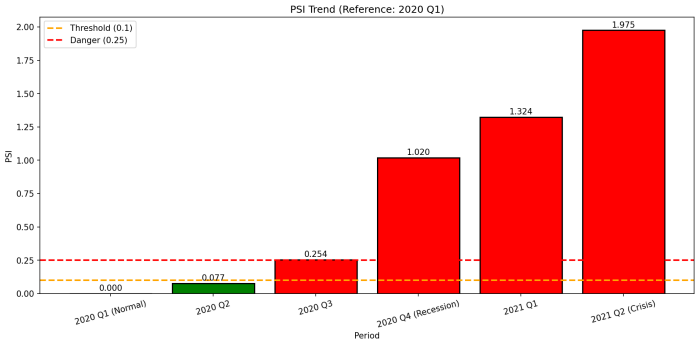

Figure 8: PSI Trend (2020 Q1 baseline)

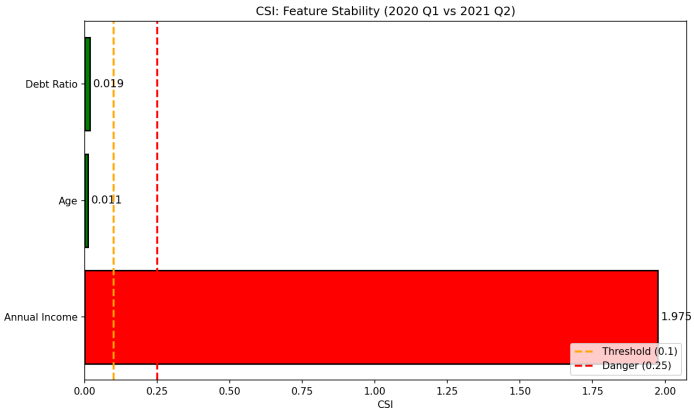

Figure 9: CSI by Feature

should include daily PSI and CSI computation, weekly model performance review, and quarterly scheduled retraining regardless of triggers. In my simulation, this early warning system detected the PSI breach 2 to 3 months before F1 collapsed, enabling proactive model updates rather than reactive firefighting.

Code and Analysis: The GitHub repository includes Jupyter notebooks with full experimental results and visualizations.