Safe Driver Prediction

Improving Pricing Fairness & Risk Control

PORTO SEGURO

Anna Esakova
Sorin Manole
Núria Muñoz
Matteo Vercelli

Exec summary

- **Problem:** Improve claim risk ranking to enable fairer pricing
- **Solution:** Stacked ensemble (XGBoost + CatBoost + LightGBM)
- **Result:** Normalized Gini ↑ from ~0.27 → ~0.29
- **Business impact:** Better separation of low vs high-risk drivers
- **Risk controls:** Bias monitoring, drift detection, human-in-the-loop

Project Goals & Product Requirements

Project Goals & Stakeholder Problem

Problem statement

Accurate claim prediction is one of the key drivers of success in the insurance business. Improving prediction quality directly enables:

- **Higher customer retention** — when good drivers are accurately identified, they no longer subsidize riskier ones
- **Pricing efficiency** — premiums better reflect actual risk, improving competitiveness
- **More reliable budget and risk planning** — fewer surprises in loss ratios and reserves
- **Stronger portfolio quality** — better separation between low- and high-risk policyholders

Our goal

Build a model that **better predicts who is likely to file a claim in the next year**, so pricing can be fairer, smarter, and more competitive.

User persona and pain points

Porto Seguro's **pricing and risk teams**' pains:

- Current models don't rank drivers accurately enough
- Safe drivers end up overpaying
- Risk signals are hidden in noisy, high-dimensional data
- Small model improvements ➔ large business impact

This benefits **good drivers**, who should finally see prices that reflect their behavior.

Product Requirements — Scope & Success

In Scope

- **Binary risk prediction**: estimating whether a claim will occur within the next policy period
- **Learning from mixed-type tabular data** (numeric, categorical, binary)
- **Optimizing for ranking quality**, consistent with insurance portfolio management objectives
- **Handling class imbalance**, where claim events are relatively rare
- **Model selection based on stability**, not just peak validation performance

Out of Scope

- Premium pricing or tariff optimization
- Real-time inference or production deployment considerations
- Regulatory explainability frameworks (e.g., formal fairness audits or compliance tooling)
- Claim severity, frequency beyond first event, or fraud detection

ML Success

- **Robustness & reliability**
  Stable Normalized Gini across cross-validation and time splits, consistent performance across customer segments
- **Strong ranking quality**
  High decile / percentile lift
- **Actionable model outputs**
  Clear and stable risk stratification, low ranking volatility, and monotonic behavior where expected
- **Operational readiness**
  Reasonable model complexity, efficient training and inference, and reproducible results across runs and random seeds.
- **Business-aligned optimization**
  Improved identification of low-risk drivers and optimization for ranking quality rather than probability calibration.

Product Requirements — Constraints & Assumptions

Constraints

- Highly imbalanced data (≈ 3,6% claims)
- Missing values encoded as -1
- Blind test set (no data labels)

Assumptions

- Train and test data come from the same distribution
- Feature groups (car, driver, region) carry useful signal
- Better ranking → better pricing outcomes

Business logic educated guesses

Probability of a claim should be predicted based on a mix of features:

- Car (type, safety features, etc)
- Individual (age, health, long working hours, etc)
- Region (quality of roads, accessibility of drivers licence, etc)

EDA &
Model Training

Dataset Structure & Key EDA Findings

- Train: ~595K rows (features + target)
- Test: ~892K rows (only features)
- 59 anonymized features grouped by **Ind** (Driver), **Reg** (Region), **Car** (Vehicle), and **Calc** (Computed).

Target Imbalance

Only 3.6% claim rate.
Strong class imbalance → accuracy is misleading.

The Missing Data Paradox

Missing data → **lower claim rate** (3.41% vs 4.54%)

Signal & Noise

Signal: Missing values are informative, not random
Noise: “Calc” features show **zero correlation** to the target → removed

Decision: GO, Despite anonymization and imbalance, ~600K rows and signal strength support a robust ML solution.

ML Framing & Metrics

Problem Framing

- Binary classification: probability of claim next year.
- Output used for **risk ranking**, not yes/no decisions.

Evaluation Metrics

- Normalized **Gini** (industry standard)
- **ROC-AUC** supports rare event ranking

Key Tradeoffs

- Recall vs Precision → prioritize ranking quality
- Accuracy vs Speed → batch scoring allows stronger models
- Performance vs Explainability → tree models balance both

Model Strategy & Experiments

Baseline & Model Families

- Started with tree-based boosting models: LightGBM, XGBoost, CatBoost
- LightGBM (CV) gave us realistic benchmark, CatBoost & XGBoost gave us strong performance

Key Improvements Tested

- Better handling of categorical variables (native vs one-hot encoder)
- Removing noisy “Calc” features
- Hyperparameter tuning and cross-validation

What Actually Moved the Metric

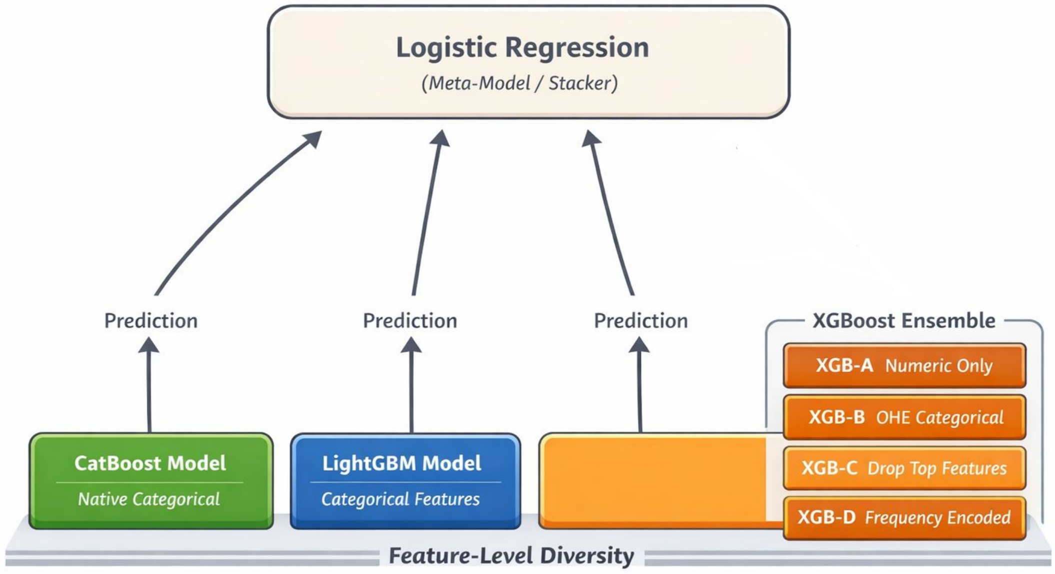
- Feature handling and data cleanup
- Use a **diverse ensemble of tuned XGBoost models** to maximize ranking performance

4. Model Experiment LOG

Experiment	Change	Result (AUC | Gini)	Decision
LightGBM Baseline	Boosted trees	0.6405 | 0.2809	Best speed/performance
LightGBM (CV)	5-fold validation	0.6341 | 0.2682	Realistic benchmark
CatBoost	Native categorical handling	**0.6416 | 0.2831**	Competitive alternative
CatBoost (CV)	Tuned + CV	0.6350 | 0.2700	Similar to LightGBM
XGBoost Baseline	Simple model with avg hyperparameters	0.6350 | 0.2700	Solid baseline
XGBoost v.2	Using OHE for all categories	0.6420 | 0.2840	Significant improvement over the baseline
XGBoost v.3	manual randomized hyperparameter search	0.6438 | 0.2876	Better accuracy
XGBoost v.4	drop calc, replace -1 with NaN	0.6460 | 0.2920	Even better accuracy
XGBoost v.5	Seed-based ensemble of 3 models	0.6457 | 0.2915	Worse because the models are too similar, and we're averaging away useful signal
XGBoost v.6	5-fold CV	0.6406 | 0.2812	Overall result is lower, perhaps we got lucky with random parameters earlier, and couldn't find a better set
XGBoost v.7	Weighted diverse ensemble of 5 models	**0.6463 | 0.2926**	Combination of 5 very different models gives the best result
Stacked model	CatBoost + LightGBM + diversified XGBoost (×4)	**Kaggle Score: 0.28168**	Each model uses **different feature encodings** → low correlation. Meta-model learns **optimal weighting + calibration**

Our Best Model

Our Best Model



Combines complementary error patterns into a single stable ranking

Expected Business Impact

- More accurate customer risk ranking
- More competitive, precise pricing decisions
- Fewer false positives ➜ lower claims volatility
- Better capital allocation and reserving

	consensus_score	cat_norm	lgb_norm	xgb_norm
ps_car_11	0.246197	0.018647	0.003209	0.716735
ps_car_11_cat	0.105906	0.014853	0.302865	0.000000
ps_car_13	0.078954	0.115453	0.098075	0.023334
ps_reg_03	0.053205	0.067551	0.080710	0.011354
ps_ind_03	0.052069	0.095573	0.048336	0.012297
ps_ind_15	0.038200	0.066044	0.037569	0.010985
ps_ind_05_cat	0.038006	0.051742	0.062277	0.000000
ps_ind_17_bin	0.032589	0.030030	0.034094	0.033644
ps_reg_01	0.031287	0.053872	0.027734	0.012256
ps_reg_02	0.028837	0.043156	0.032931	0.010424

The "Black Box" Risk Analysis

The Consensus Model Reality

- **Dominant Signal:** The model is heavily profiling the **Vehicle**. The top 3 features (`ps_car_11`, `ps_car_11_cat`, `ps_car_13`) make up **~43%** of the total weight.

- **Ethical Red Flag:**
  ps_reg_03 (region) is the #4 most important feature.

The Danger: **Digital Redlining**. Pricing based on "Region" + "Car Model" is a strong proxy for Socioeconomic Status, not just driving skill.

Mitigation:

Mandatory **Disparate Impact Testing** across regional clusters before launch.

Resilience & The Safety Net

Technical Robustness (The Ensemble Advantage)

- **The Component Failure:** XGBoost was critically fragile (71% reliance on `ps_car_11`). *The Fix:* The Consensus Model **dilutes this risk**. **The top feature dependency dropped from 71% → 24%.**
- **Benefit:** If the `ps_car_11` data feed breaks, LightGBM and CatBoost (which rely on other features) stabilize the output.

Remaining Risk

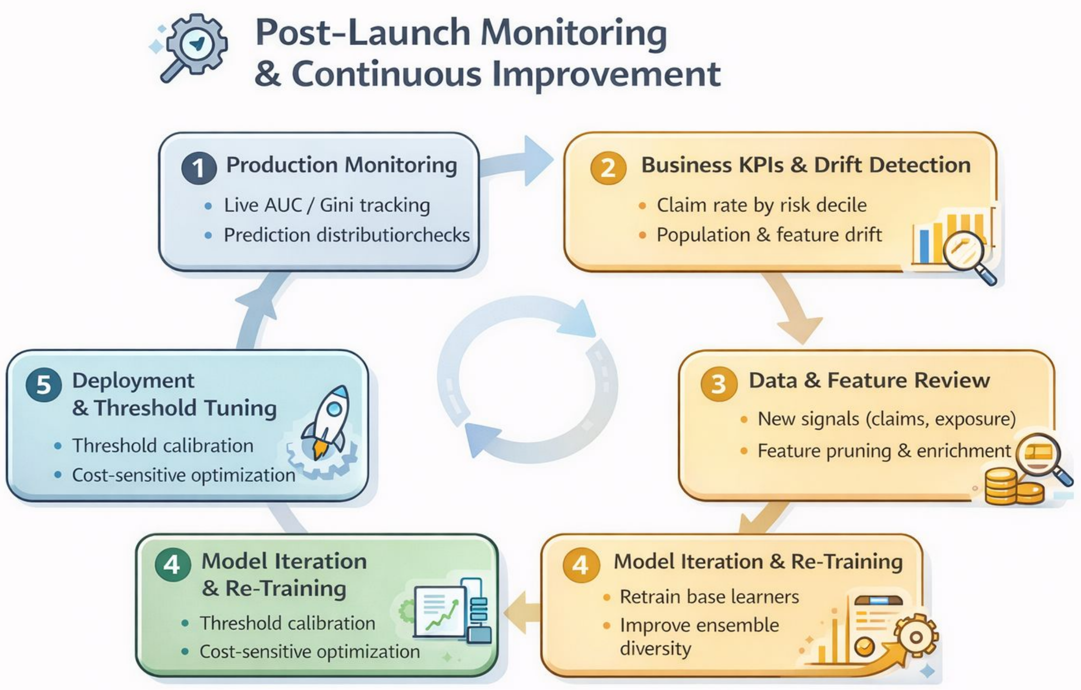
- **"New Car" Cold Start:** Since ~35% of the score depends on Car Model (`car_11` + `car_11_cat`), new vehicles may be mispriced.

The Safety Net (Human-in-the-Loop)

- **Safety Net:** Trigger manual review if the model detects a "New/Unknown" Vehicle Code.

Post-Launch Map & Iteration Roadmap

- Monitor performance and drift
- Validate with real outcomes
- Retrain and recalibrate regularly
- Iterate features and ensemble



Thank you

PORTO SEGURO