1. Introduction and Problem Understanding

This report documents our approach to the customer churn prediction competition, where the objective was to predict which users of a music streaming service would cancel their subscription within a 10-day window following November 20, 2018. The journey from initial exploration to final submission taught us critical lessons about handling imbalanced data, feature engineering philosophy, and the importance of reframing classification problems.

Our first encounter with the data revealed a fundamental challenge that would shape our entire approach: severe class imbalance. With only approximately 3% of users churning in the training set, a naive classifier predicting "no churn" for everyone would achieve 97% accuracy while being completely useless. This immediately told me that accuracy was the wrong metric to optimize.

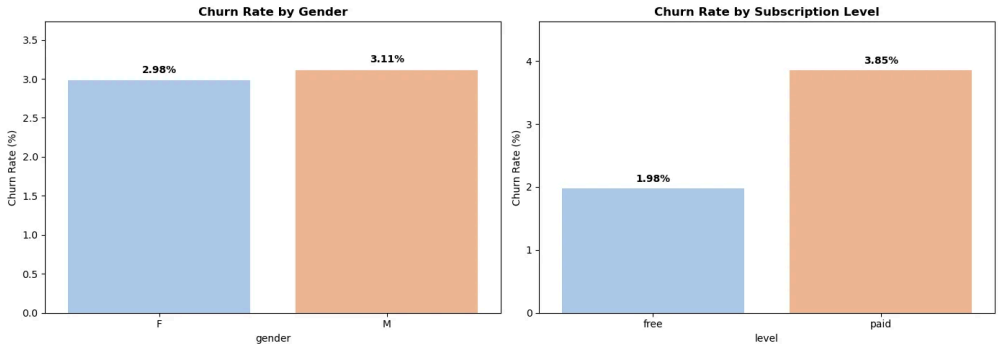

Figure 1: Churn rates by demographic segments (~3% overall)

A crucial discovery came when we analyzed the test set structure: it was artificially rebalanced to approximately 50% churn rate. This distribution shift between training and test data meant that probability calibration would be unreliable. We realized we needed to reframe this as a ranking problem rather than a classification problem, the goal became learning a scoring function that correctly orders users by their churn risk, regardless of absolute probability values.

Three constraints shaped the methodology throughout the competition:

- **Temporal integrity:** All features must be computed using only data before the prediction cutoff to avoid data leakage
- **Leakage prevention:** The "Cancellation Confirmation" page directly indicates churn and must be excluded from features
- **Evaluation metric:** AUC (Area Under ROC Curve) measures ranking quality, making it the ideal metric for this reframed problem

2. Feature Engineering

The most impactful insight of our approach came from thinking about what truly predicts churn. Raw activity counts are misleading because they're biased by user tenure, a long-time user who stopped engaging two weeks ago will still have higher total song counts than an active new user. We needed features that capture *behavioral change* rather than cumulative volume.

A key realization was that raw counts are biased by user tenure, a long-time user who stopped engaging will still have higher totals than an active new user. We focused on normalized rate features that capture behavior intensity regardless of account age. The most predictive feature turned out to be *std_song_length* (standard deviation of song length), which captures listening variability, users with consistent patterns may be more engaged than those with erratic behavior. Other top

features like *add_friend_rate* and *advert_rate* measure social investment and frustration signals normalized by activity volume.

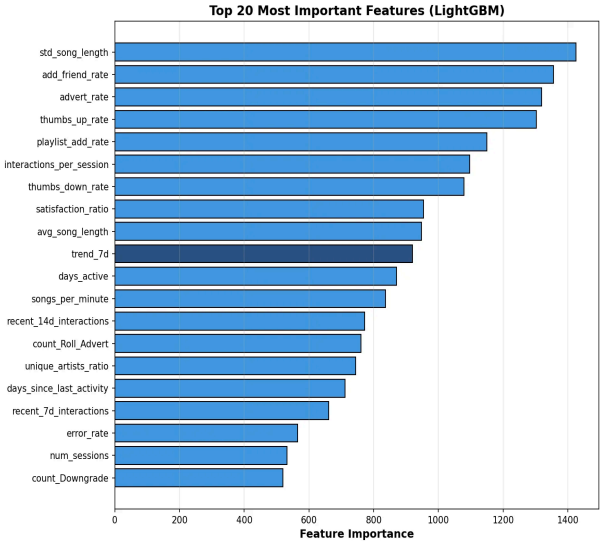

Figure 2: Feature importance ranking from LightGBM model

Our final feature set comprised six conceptual categories, each designed to capture different aspects of user behavior and engagement:

- **Engagement intensity:** Standard deviation of song length, interactions per session, songs per minute, capturing how users consume content
- **Social investment:** Add friend rate, playlist addition rate, indicating emotional investment in the platform
- **Frustration signals:** Advertisement rate, thumbs down rate, error rate, negative experiences that erode satisfaction
- **Temporal dynamics:** Days active, days since last activity, 7-day and 14-day trend ratios
- **Satisfaction metrics:** Thumbs up/down ratio, unique artists ratio, average song length
- **Composite risk score:** A domain-driven heuristic combining cancel intent, declining trends, error rates, and inactivity

3. Model Development and Training Strategy

Rather than training on a single snapshot, we created three distinct training windows (October 20, October 30, November 10) with their corresponding churn labels. This approach tripled the training data without introducing synthetic noise, allowing the model to learn churn patterns at different stages of the user lifecycle. This was far more effective than SMOTE or other synthetic oversampling techniques, which we found blurred decision boundaries and hurt performance.

We experimented with multiple model families: Logistic Regression and LDA as linear baselines, plus gradient boosting methods (LightGBM, CatBoost, XGBoost). Using Optuna for hyperparameter optimization with 5-fold stratified cross-validation, we found that LGBM with regularization achieved a performance at 0.7736 CV AUC, and a private score of 0,65198 for the best parameters.

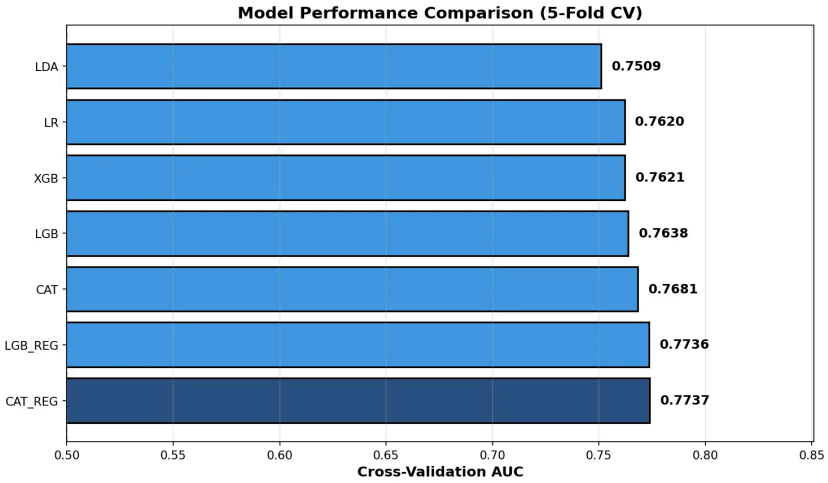

Figure 3: Model performance comparison (5-fold cross-validation AUC)

Key hyperparameter insights emerged from tuning: low learning rates (0.005-0.02), moderate tree depth (4-6), and strong regularization (L2 leaf regularization around 4-5) were crucial for preventing overfitting to the small minority class. The "micro-tuning" philosophy, conservative parameters with heavy regularization, proved more effective than aggressive optimization.

Our ensemble approach used rank averaging instead of probability averaging. Each model's predictions are converted to ranks (1 to N), then averaged across models. This technique is robust to calibration differences between models, different algorithms produce probability distributions with varying confidence levels, but their *rankings* tend to be more consistent. Since AUC only depends on ranking quality, this approach directly optimizes our evaluation metric.

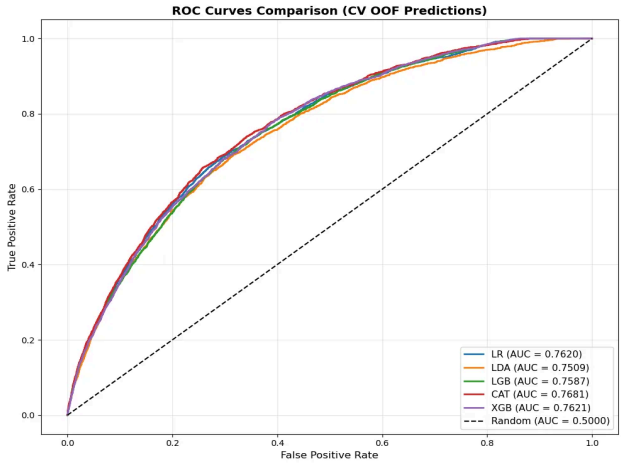

Figure 4: ROC curves showing similar discrimination ability across models

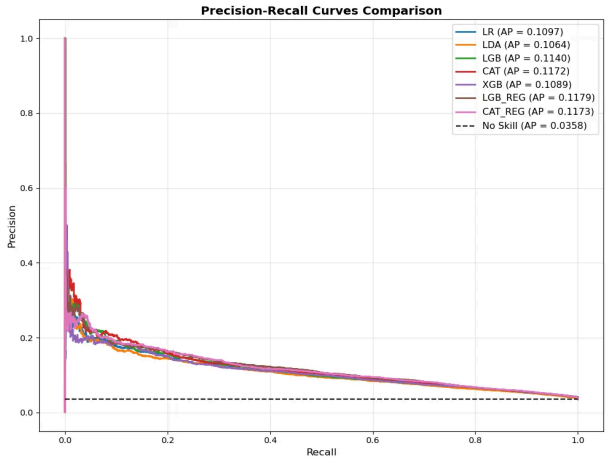

Figure 5: Precision-Recall curves—critical for evaluating imbalanced classification

While ROC-AUC is our competition metric, we also monitored the Precision-Recall curve, a more informative evaluation for severely imbalanced datasets. Unlike ROC curves, which can appear optimistic when the negative class dominates, Precision-Recall curves focus exclusively on the minority class performance. With only ~3.5% churners, this visualization reveals the true difficulty of the task.

The Average Precision (AP) scores around 0.11-0.12 may seem low, but they represent roughly 3x improvement over the random baseline (AP ≈ 0.036, equal to the churn rate). This confirms that our models capture meaningful churn signals despite the extreme class imbalance. The regularized boosting models (LGB_REG, CAT_REG) achieve the highest AP scores, validating the micro-tuning approach.

4. Results And Conclusion

Our final submission achieved a private leaderboard AUC of 0.64308 *(NOT THE BEST ONE WICH IS 0,65198)*. Cross-validation scores ranged from 0.75-0.77, with the gap to leaderboard performance reflecting the distribution shift between training and test sets. The threshold optimization for the artificially balanced test set (targeting ~50% positive predictions) was crucial for maximizing leaderboard performance.

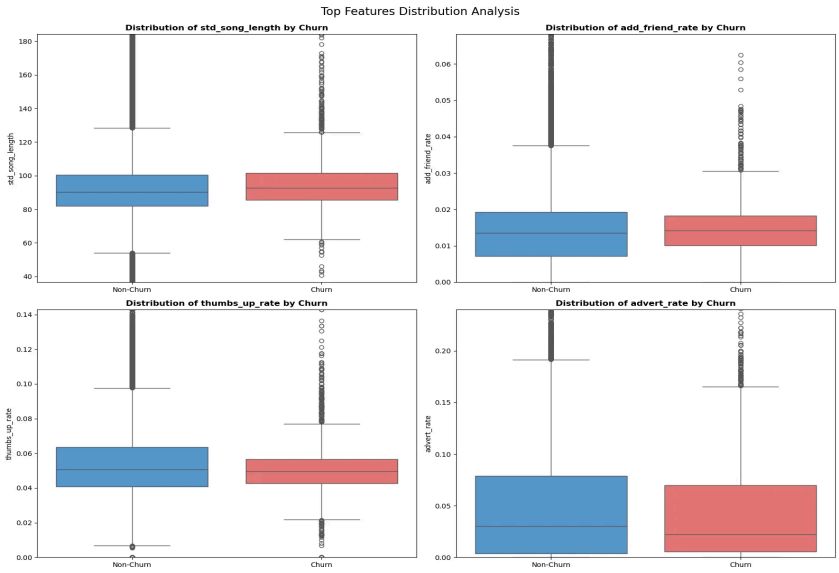

Figure 6: Distribution analysis of top predictive features by churn status

What Worked : The *Rate-based features* like std_song_length, add_friend_rate, and advert_rate dominated feature importance by normalizing behavior by activity volume. The *Multi-window training* tripled data volume without synthetic noise, significantly improving generalization. The *Rank averaging* provided calibration-robust ensembling, directly optimizing for the ranking metric. The *feature cleaning/selection* removed only perfect duplicates and metadata, preserving subtle predictive signals

What Didn't Work : The *SMOTE oversampling* generated synthetic churners that blurred decision boundaries. The *Pseudo-labeling* on high-confidence test predictions introduced bias without improvement. The *Aggressive feature dropping* based on correlation removed features with subtle but valuable signals

To conclude, this competition reinforced that problem formulation matters more than model complexity. Reframing churn prediction as a ranking problem rather than classification, and focusing on normalized rate features rather than absolute counts, were more impactful than any hyperparameter tuning. The dominance of *std_song_length* as the top feature was surprising, it suggests that listening variability patterns reveal engagement quality better than volume metrics. Combined with social investment signals (add_friend_rate) and frustration indicators (advert_rate), these rate-based features validated that domain-driven feature engineering remains the highest-leverage activity in machine learning competitions.

Future improvements could explore time series modeling for seasonal patterns, user clustering for segment-specific models, and explicit feature interactions between key behavioral signals.