Predicting Student Exam Scores Using Tabular Machine Learning

Student 1
Department
University Name
City, Country
email@domain.com

Student 2
Department
University Name
City, Country
email@domain.com

Student 3
Department
University Name
City, Country
email@domain.com

***Abstract*—This project addresses the Kaggle Playground Series S6E1 competition: predicting student exam scores from demographic, academic, and behavioral features. Using a dataset of 630,000 training samples with 12 original features, we developed a comprehensive machine learning pipeline including feature engineering (expanding to 44 features), multiple model comparison, and ensemble optimization. We evaluated Ridge Regression, Random Forest, and gradient boosting methods (LightGBM, XGBoost, CatBoost). Our best solution—a weighted ensemble of CatBoost (62%), LightGBM (33%), and XGBoost (5%)—achieved a cross-validation RMSE of 8.7568. Analysis revealed that study hours (correlation r=0.76) and engineered interaction features were the strongest predictors. We discuss feature importance, error patterns, and limitations of the synthetic dataset.**

***Index Terms*—regression, tabular data, machine learning, gradient boosting, cross-validation, RMSE**

I. Introduction

Academic performance prediction is a significant machine learning application with implications for personalized learning and early intervention systems. In this work, we address the Kaggle Playground Series S6E1 competition, where the objective is to predict student exam scores from behavioral and contextual features including study hours, class attendance, sleep patterns, and study methodology.

The dataset comprises 630,000 training samples and 270,000 test samples, generated synthetically from a deep learning model trained on real exam score data. This provides a controlled environment to compare various machine learning approaches.

The main contributions of this paper are:

- A comprehensive exploratory analysis revealing that `study_hours` (r=0.76) is the strongest predictor of exam scores.
- Feature engineering expanding 12 original features to 44 derived features including interactions, target encoding, and domain-specific formulas.
- Comparison of 7 regression models from Ridge to gradient boosting ensembles.
- Achievement of 8.7568 CV RMSE using an optimized weighted ensemble of CatBoost, LightGBM, and XGBoost.
- Analysis of feature importance and prediction error patterns.

II. Technical Background

This task is a supervised regression problem. Given an input feature vector $x \in \mathbb{R}^d$, we aim to predict a continuous target value y, representing the exam score.

A. Root Mean Squared Error (RMSE)

The primary evaluation metric is Root Mean Squared Error:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{1}$$

Lower RMSE indicates better predictive performance.

B. Baseline Models

We consider standard regression baselines such as Linear Regression and Ridge Regression, which are simple and interpretable.

C. Gradient Boosted Decision Trees

We also evaluate gradient boosting models (e.g., XGBoost, LightGBM, CatBoost), which often perform strongly on structured/tabular datasets by combining many weak decision tree learners into a robust model.

III. Dataset Description and Preprocessing

The dataset originates from the Kaggle Playground Series S6E1 competition, consisting of synthetically generated student data. The training set contains 630,000 samples with the target label `exam_score`, while the test set contains 270,000 samples requiring predictions.

A. Feature Overview

The dataset contains 12 original features (excluding `id`), split into:

- **Numerical (4)**: age (17–24), study_hours (0.08–7.91), class_attendance (40.6–99.4%), sleep_hours (4.1–9.9)
- **Categorical (7)**: gender, course (B.Tech, B.Sc, etc.), internet_access, sleep_quality (good/average/poor), study_method, facility_rating, exam_difficulty

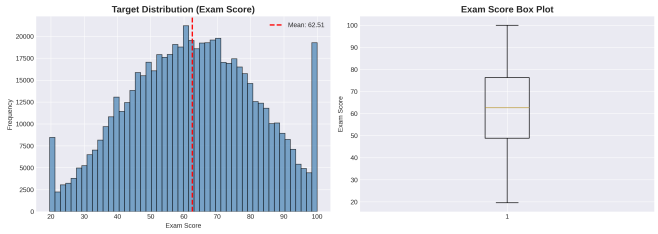

Fig. 1. Distribution of the target variable `exam_score` (histogram and box plot).

The target variable `exam_score` has the following distribution:

- Mean: 62.51, Standard Deviation: 18.92
- Range: 19.60 – 100.00
- Skewness: -0.048 (approximately symmetric)

B. Data Cleaning

We performed the following preprocessing steps:

- Verified **no missing values** in both training and test sets.
- Applied label encoding for categorical variables.
- Implemented CV-safe target encoding to prevent data leakage.
- Added frequency encoding for categorical features.

C. Exploratory Data Analysis (EDA)

EDA revealed important relationships between features and exam scores:

- **study_hours** shows the strongest correlation (r = 0.762) with exam score.
- **class_attendance** has moderate positive correlation (r = 0.361).
- **sleep_hours** exhibits weak positive correlation (r = 0.167).
- **age** shows negligible correlation (r = 0.011).

Figure 2 summarizes linear correlations among numerical variables, while Figures 3 and 4 illustrate feature-target relationships for numerical and categorical variables.
Categorical variables like `sleep_quality`, `study_method`, and `facility_rating` showed meaningful differences in score distributions across categories.

IV. Methodology

Our modeling pipeline follows a systematic approach from baseline models to advanced ensembles.

A. Validation Strategy

To ensure reliable performance estimation and prevent overfitting:

- **5-fold cross-validation** with shuffle (random seed = 42)
- Consistent fold splits across all models for fair comparison
- Out-of-fold (OOF) predictions stored for ensemble optimization

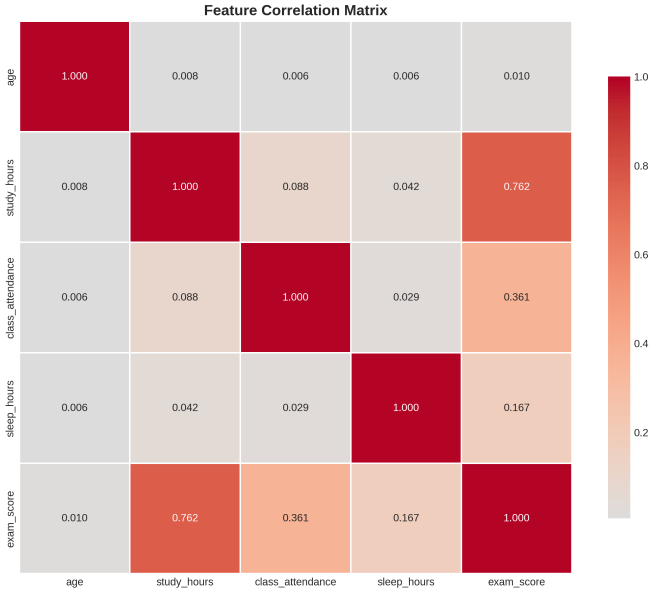

Fig. 2. Correlation matrix of numerical features and the target.

B. Feature Engineering

We expanded from 12 original features to **44 engineered features**:

- **Interaction features**: study_hours × class_attendance, study_hours × sleep_hours, etc.
- **Ratio features**: attendance_per_hour, sleep_per_study
- **Polynomial features**: squared and cubed terms for key numerical variables
- **Binary flags**: is_low_sleep (<6h), is_high_attendance (>80%), is_high_study (>6h)
- **Domain formula**: $6.0 \times$study_hours$+0.35 \times$attendance$+$ $1.5 \times$ sleep_hours
- **Target encoding**: CV-safe mean encoding for all categorical features
- **Frequency encoding**: Category frequency as additional features

C. Models Evaluated

We evaluated models of increasing complexity:

Baseline Models:

- Ridge Regression ($\alpha = 10.0$) with standardized features
- Random Forest (100 trees, max_depth=15, min_samples_leaf=10)

Gradient Boosting Models:

- LightGBM (learning_rate=0.05, num_leaves=31, early stopping at 100 rounds)
- XGBoost (learning_rate=0.05, max_depth=6, tree_method=hist)
- CatBoost (learning_rate=0.05, depth=6, l2_leaf_reg=3)

Ensemble Methods:

- Simple average of LightGBM, XGBoost, CatBoost predictions

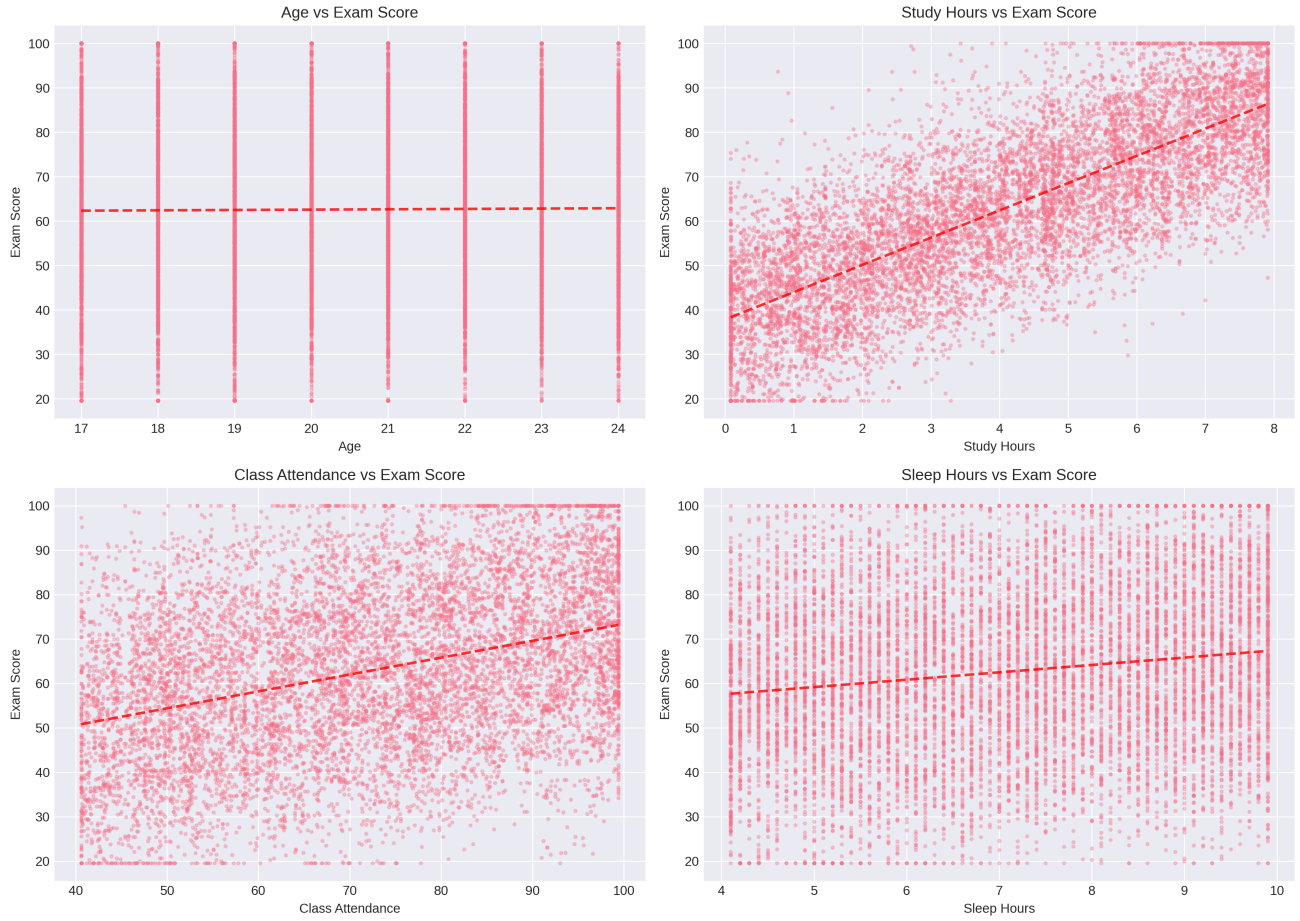

Fig. 3. Scatter plots of numerical features versus `exam_score` with linear trend lines (sampled for visualization).

- Weighted ensemble with Nelder-Mead optimized weights on OOF predictions

V. Experiments and Results

A. Experimental Setup

All experiments used identical preprocessing, feature engineering, and 5-fold cross-validation splits. Performance is measured by mean CV RMSE (Root Mean Squared Error), with standard deviation indicating stability across folds.

B. Model Comparison

Table I summarizes the performance of each method, sorted by CV RMSE.

TABLE I
Model Performance Comparison (Lower RMSE is Better)

Model	CV RMSE	Std
Ensemble (Optimized)	**8.7568**	–
CatBoost	8.7618	±0.0101
LightGBM	8.7724	±0.0089
Ensemble (Simple)	8.7730	–
Ridge Regression	8.8887	±0.0105
Random Forest	8.9095	±0.0109
XGBoost	8.9271	±0.1928

Key observations:

- Gradient boosting models outperform linear and tree-based baselines by ∼0.12–0.15 RMSE.
- CatBoost achieved the best single-model performance (8.7618).
- The weighted ensemble provides marginal improvement (∼0.005) over the best single model.
- XGBoost showed higher variance across folds (±0.19) compared to other models.

C. Ensemble Optimization

The weighted ensemble was optimized using Nelder-Mead minimization on out-of-fold predictions:

- **CatBoost**: 61.98% weight
- **LightGBM**: 33.16% weight
- **XGBoost**: 4.86% weight

The low weight assigned to XGBoost reflects its higher variance and slightly worse performance.

D. Feature Importance

Feature importance analysis (based on the best single model, CatBoost) revealed the top predictors:

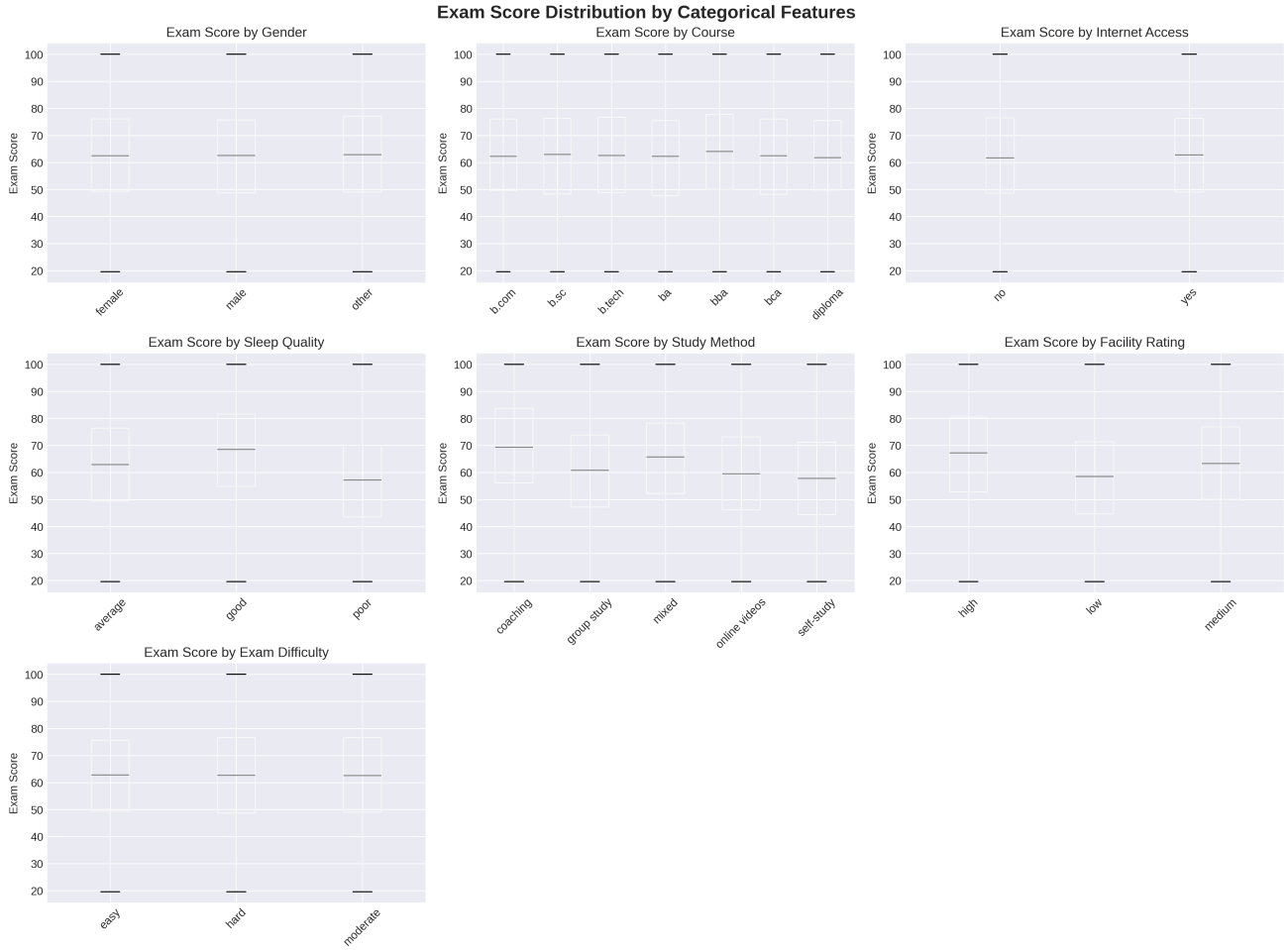

Fig. 4. Score distributions across categorical features. Categories with higher `sleep_quality`, `study_method`, and `facility_rating` tend to show higher median scores.

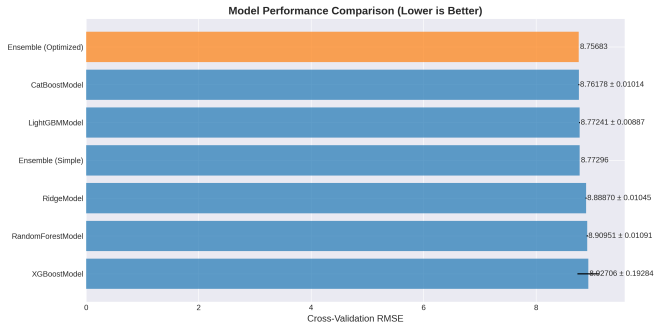

Fig. 5. Cross-validation RMSE comparison across all models and ensembles (lower is better).

TABLE II
TOP 10 MOST IMPORTANT FEATURES (CATBOOST)

Feature	Importance
formula_score	60.78
sleep_quality_target_enc	5.99
study_method_target_enc	4.95
facility_rating_target_enc	4.10
study_attendance	3.48
sleep_quality	2.70
study_method	2.62
sleep_quality_freq	2.21
facility_rating	2.14
study_method_freq	1.75

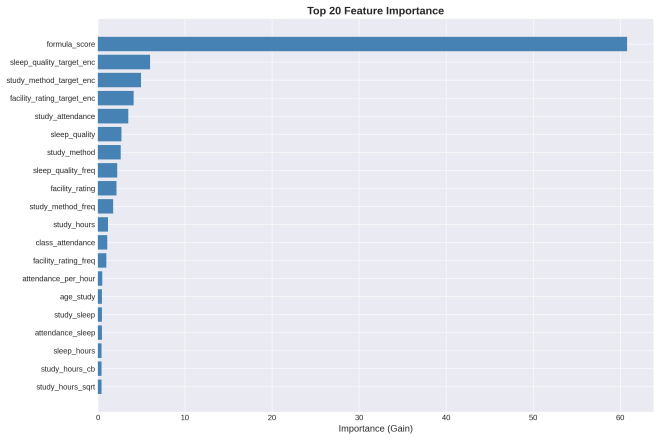

Fig. 6. Feature importance for the best single model (CatBoost). The engineered `formula_score` and category encodings dominate.

The engineered `formula_score` feature dominates, followed by encodings of `sleep_quality` and `study_method`, confirming that feature engineering

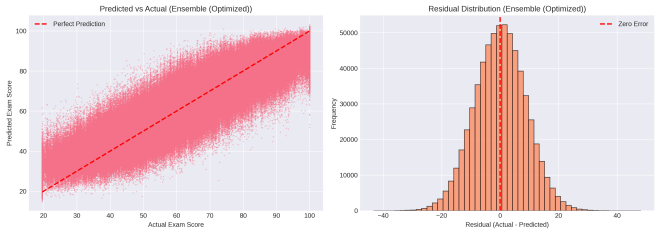

Fig. 7. Residual analysis for the best model (optimized ensemble): predicted vs. actual scores and residual distribution.

significantly improved predictive power.

E. Residual Analysis

For the best model (Weighted Ensemble), residual statistics on the training set:

- Mean residual: 0.0415 (near-zero bias)
- Residual std: 8.7567 (consistent with CV RMSE)
- Range: [-43.35, +48.15] (some large errors remain)

VI. Discussion

A. Model Performance Analysis

The results demonstrate clear performance hierarchy:

1) **Gradient boosting models** (CatBoost, LightGBM) achieved the best performance (∼8.76 RMSE), improving over baselines by 1.3–1.7%.
2) **Ensemble methods** provided marginal but consistent gains, with the weighted ensemble achieving the best overall RMSE of 8.7568.
3) **CatBoost** slightly outperformed LightGBM as a single model, likely due to its sophisticated handling of categorical features.
4) **XGBoost** underperformed expectations with higher variance, potentially due to suboptimal hyperparameters or sensitivity to the feature set.

B. Feature Engineering Impact

The engineered `formula_score` feature, based on domain knowledge from competition discussions, became the single most important predictor. This highlights the value of incorporating domain expertise and exploring competition forums for insights.

Target encoding and interaction features (especially `study_attendance`) contributed significantly, demonstrating that careful feature engineering can outweigh model selection in tabular competitions.

C. Error Analysis

Residual analysis revealed:

- Near-zero mean residual (0.0415) indicates minimal systematic bias.
- Residual standard deviation (8.7567) aligns with the cross-validation RMSE, indicating consistent error magnitude.
- Residual range of [-43.35, +48.15] suggests difficulty predicting extreme scores.
- Largest errors likely occur for students with unusual combinations of features.

D. Limitations

Several limitations apply to this work:

- **Synthetic data**: The dataset was generated from a deep learning model, so relationships may not perfectly reflect real-world patterns.
- **Computational constraints**: More extensive hyperparameter tuning and neural network approaches were not explored due to time constraints.
- **Leakage potential**: While CV-safe target encoding was used, additional care should be taken in production settings.
- **Generalization**: Results are specific to this synthetic dataset and may not transfer to real student data.

VII. Conclusion

This project addressed the Kaggle Playground Series S6E1 challenge of predicting student exam scores from tabular demographic and behavioral data.

A. Summary of Contributions

- Developed a comprehensive feature engineering pipeline expanding 12 original features to 44 engineered features.
- Compared 7 modeling approaches from linear baselines to ensemble methods.
- Achieved a final CV RMSE of **8.7568** using a weighted ensemble of CatBoost (62%), LightGBM (33%), and XGBoost (5%).
- Identified `study_hours` and its interactions as the most predictive features.

B. Key Findings

1) **Feature engineering matters**: The engineered `formula_score` and interaction features contributed more than model selection.
2) **Gradient boosting dominates**: CatBoost and LightGBM significantly outperformed traditional models.
3) **Ensemble provides marginal gains**: Weighted blending improved over single models by ∼0.005 RMSE.
4) **Study behavior is key**: Study hours and class attendance are the strongest predictors of exam performance.

C. Future Work

Potential improvements include:

- Systematic hyperparameter optimization using Optuna or Bayesian methods
- Neural network approaches (TabNet, deep embeddings)
- Stacking ensembles with meta-learners
- Incorporating the original (non-synthetic) dataset for additional training signal

APPENDIX

All experiments used a fixed random seed (SEED=42) for dataset splitting and cross-validation. Implementation was done in Python using:

- pandas, numpy for data manipulation
- scikit-learn for preprocessing and baseline models
- LightGBM, XGBoost, CatBoost for gradient boosting
- scipy for ensemble weight optimization

Tables III, IV, and V report hyperparameters for each gradient boosting model.

TABLE III
LIGHTGBM HYPERPARAMETERS

Parameter	Value
learning_rate	0.05
num_leaves	31
feature_fraction	0.9
bagging_fraction	0.8
min_child_samples	20
reg_alpha	0.1
reg_lambda	0.1
early_stopping_rounds	100

TABLE IV
CATBOOST HYPERPARAMETERS

Parameter	Value
learning_rate	0.05
depth	6
l2_leaf_reg	3
iterations	10000
early_stopping_rounds	100

TABLE V
XGBOOST HYPERPARAMETERS

Parameter	Value
learning_rate	0.05
max_depth	6
subsample	0.8
colsample_bytree	0.9
reg_alpha	0.1
reg_lambda	0.1
tree_method	hist
early_stopping_rounds	100

The final weighted ensemble used the following optimized weights:

- CatBoost: 61.98%
- LightGBM: 33.16%
- XGBoost: 4.86%