Comparing Performance Gains from Hyperparameter and Threshold Optimization Using Hyperopt, Optuna, and Scikit-learn's TunedThresholdClassifierCV for Three Gradient Boosting Algorithms on Binary Classification

Abstract

This project aims to compare the performance of three gradient boosting algorithms on a medical diagnosis binary classification task. The focus is on prediction performance resulting from hyperparameter and threshold optimization, rather than speed or computational cost.

Project structure

1. Importing and Installing Libraries
2. Loading and Previewing the Dataset
3. Defining Helper Functions
4. Exploratory Data Analysis
    - Summary Statistics
    - Distribution Analysis
    - Missing Values Analysis
    - Correlation Analysis
    - Outcome Analysis
5. Data Processing
    - Model-Based Imputations
    - Synthetic Sampling
    - Feature Selection
6. Training Gradient Boosting Classifiers
    - Building Baseline Models
    - Optimizing Hyperparameters
    - Optimizing the Classification Threshold
7. Performance Analysis

Data

The data used in this analysis comes from the Pima Indians Diabetes Database. The main challenges with this dataset include:

- Small number of records
- Large number of missing measurements
- Class imbalance

Techniques employed to address these issues include model-based imputations, synthetic generation of samples for the minority class, and feature selection to refine performance metrics.

The dataset consists of the following columns:

1. **Pregnancies**: Number of times pregnant
2. **Glucose**: Plasma glucose concentration
3. **BloodPressure**: Diastolic blood pressure (mm Hg)
4. **SkinThickness**: Triceps skin fold thickness (mm)
5. **Insulin**: 2-Hour serum insulin (mu U/ml)

6. **BMI**: Body mass index (weight in kg/(height in m)^2)
7. **DiabetesPedigreeFunction**: Diabetes pedigree function (a function which scores likelihood of diabetes based on family history)
8. **Age**: Age in years
9. **Outcome**: Class variable (0 or 1, indicating whether the patient has diabetes)

Methods

The following is a detailed description of the methods used in this project.

1. Model-based Imputations

Gradient boosting regression was employed to predict missing values. This model-based approach introduces additional complexity and requires more computational resources, which may not be suitable for very large datasets. However, given the manageable size of the current dataset, LightGBM with hyperparameter optimization via Hyperopt was used for imputation. Proper treatment of missing values significantly impacts the training and optimization of the final models, making it an essential component of the production pipeline.

2. Feature Scaling

While tree-based models are robust to differences in feature scales due to their automatic feature selection and splitting based on information gain, properly scaled features can enhance performance and facilitate testing with other models that require feature scaling.

3. Data Augmentation

To address the underrepresentation of the minority class in the training data, the Adaptive Synthetic Sampling (ADASYN) technique was utilized to generate new samples. ADASYN creates synthetic samples based on the harder-to-classify instances, leading to more effective learning.

4. Feature Selection

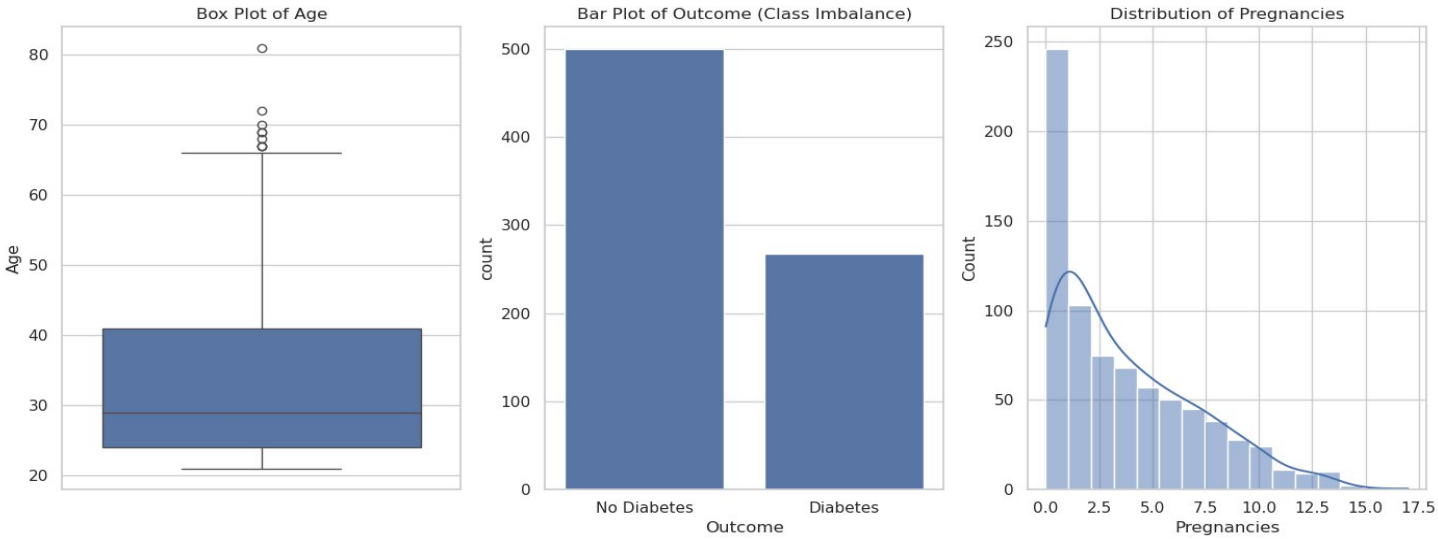
Although tree-based gradient boosting models inherently assess feature importance, explicit feature selection can still enhance optimization results. The `SelectKBest` feature selector from scikit-learn, using the `f_classif` scoring function (an ANOVA F-value method), was applied to determine feature relevance. The number of features (K) to be selected was determined by monitoring the performance of classifiers and optimizers with different K values.

5. Class Weighting

To further address class imbalance, the `scale_pos_weight` parameter was used to adjust the weights of positive (minority) class instances during training. This parameter was manually calculated for baseline models and automatically determined during optimization. The combination of ADASYN and class weighting synergistically improved performance, particularly recall for the minority class.

Insights from EDA

Summary statistics indicate that the subjects' ages range from 21 to 81 years, with a median age of 29. The target variable shows a 53.6% positive-to-negative ratio. The number of pregnancies ranges up to 17, with a median of 3. Features such as insulin, glucose, BMI, and blood pressure have minimum values of zero, suggesting zero-imputation for missing measurements.



Age distribution (left), class imbalance (middle) and distribution of pregnancies (right).

Correlation analysis reveals notable positive correlations of Outcome with Glucose(0.47), BMI(0.29), Age(0.24), and Pregnancies(0.22). There are moderate to strong correlations between Insulin and SkinThickness (0.44), SkinThickness and BMI(0.39), Insulin and BMI (0.20). These correlations were used to guide the feature selection process for model-based imputations.

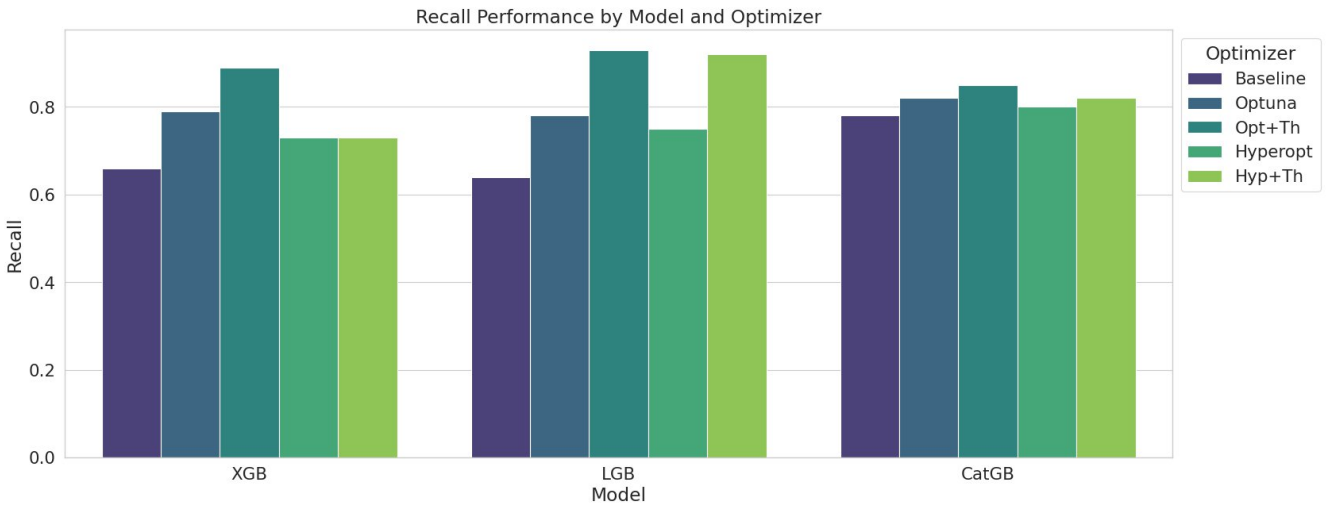
Training and testing gradient boosting classifiers

To evaluate the performance of the three gradient boosting algorithms, a baseline model was trained and tested without optimization for each algorithm. Then, hyperparameters were optimized using Optuna and Hyperopt with cross-validation. After optimization, new models were trained with the optimized hyperparameters and tested on the test set. Additionally, to balance the trade-off between precision and recall, the decision threshold was optimized using the TunedThresholdClassifierCV with an F1 scorer.

Results

Analysis of Recall Performance

For this binary classification task in medical diagnosis, the most critical metric is recall, representing the rate of true positive predictions. Recall scores ranged from approximately 0.64 to 0.89, with some models and optimizers achieving higher recall.



The grouped bar graph shows the recall performance of the three gradient boosting models (XGBoost, LightGBM, and CatBoost) across different optimization methods.

Key Observations:

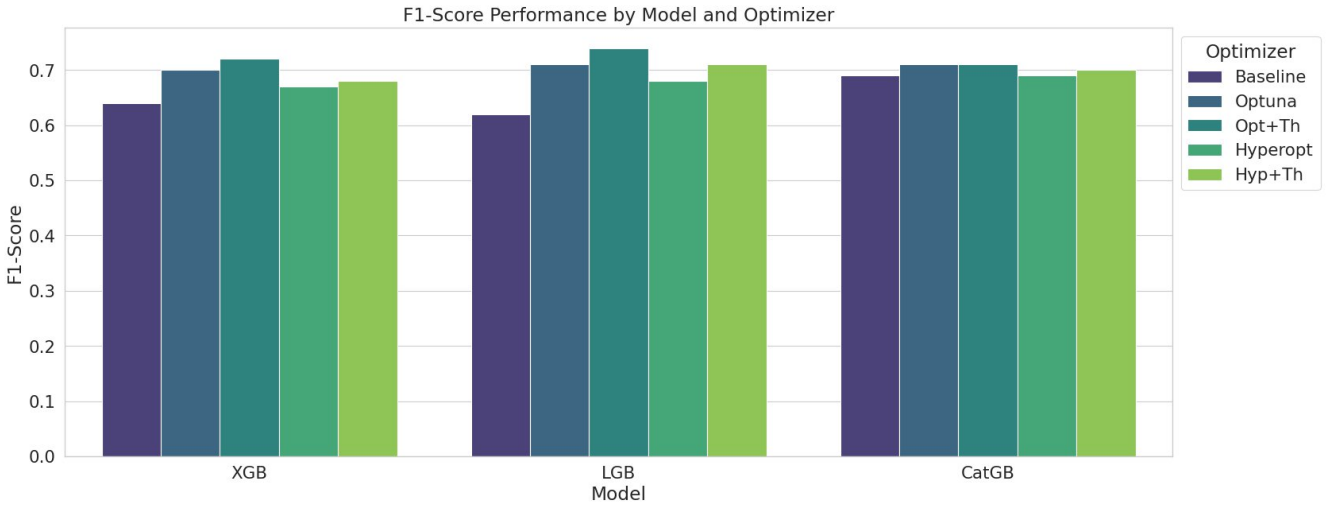
1. **XGBoost (XGB)**
   - The recall score improves significantly with the Optuna and Opt+Th optimizations compared to the baseline.
   - The highest recall is achieved with the Opt+Th optimizer.

2. **LightGBM (LGB)**
   - Similar to XGBoost, LightGBM shows improved recall scores with optimizations.
   - Opt+Th also provides the highest recall for LightGBM.

3. **CatBoost (CatGB)**
   - CatBoost exhibits strong recall performance even with the baseline model.
   - The highest recall for CatBoost is achieved with the Opt+Th optimizer.

Analysis of F1-Score Performance

The F1-score is a balance between precision and recall. It provides a single metric to evaluate the performance of a classifier, especially useful for imbalanced datasets. The F1 scores range from around 0.62 to 0.72.



The grouped bar graph shows the F1-score performance of the three gradient boosting models (XGBoost, LightGBM, and CatBoost) across different optimization methods.

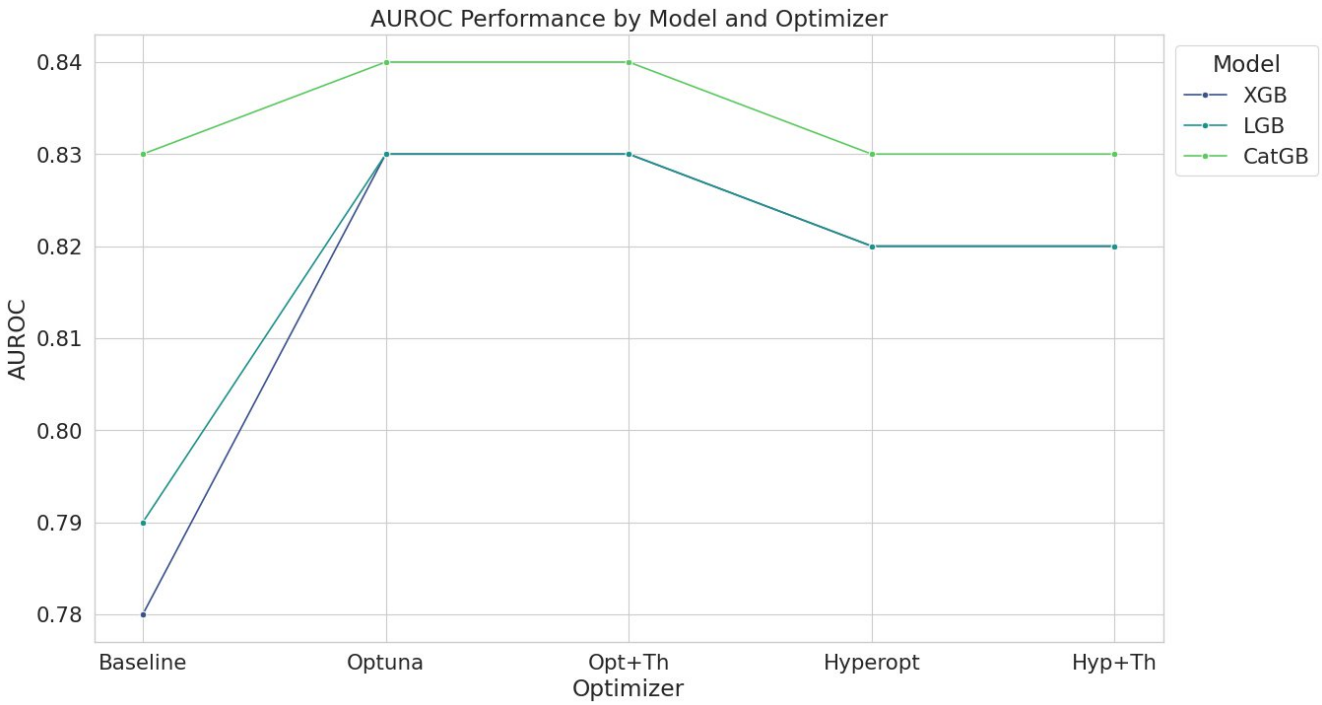
Key Observations:

1. **XGBoost (XGB)**
    - The F1-score improves with optuna and Opt+Th optimizations compared to the baseline.
    - The highest F1-score is achieved with the Opt+Th optimizer.

2. **LightGBM (LGB)**
    - LightGBM shows improvement in F1-score with Optuna and opt+Th optimizations.
    - The Opt+Th optimizer yields the highest F1-score.

3. **CatBoost (CatGB)**
    - CatBoost demonstrates strong F1-score performance even with the baseline model.
    - The highest F1-score for CatBoost is achieved with the Opt+Th optimizer.

AUROC (Area Under the Receiver Operating Characteristic Curve)



The line plot shows the AUROC variation across different optimization methods for the three gradient boosting models.

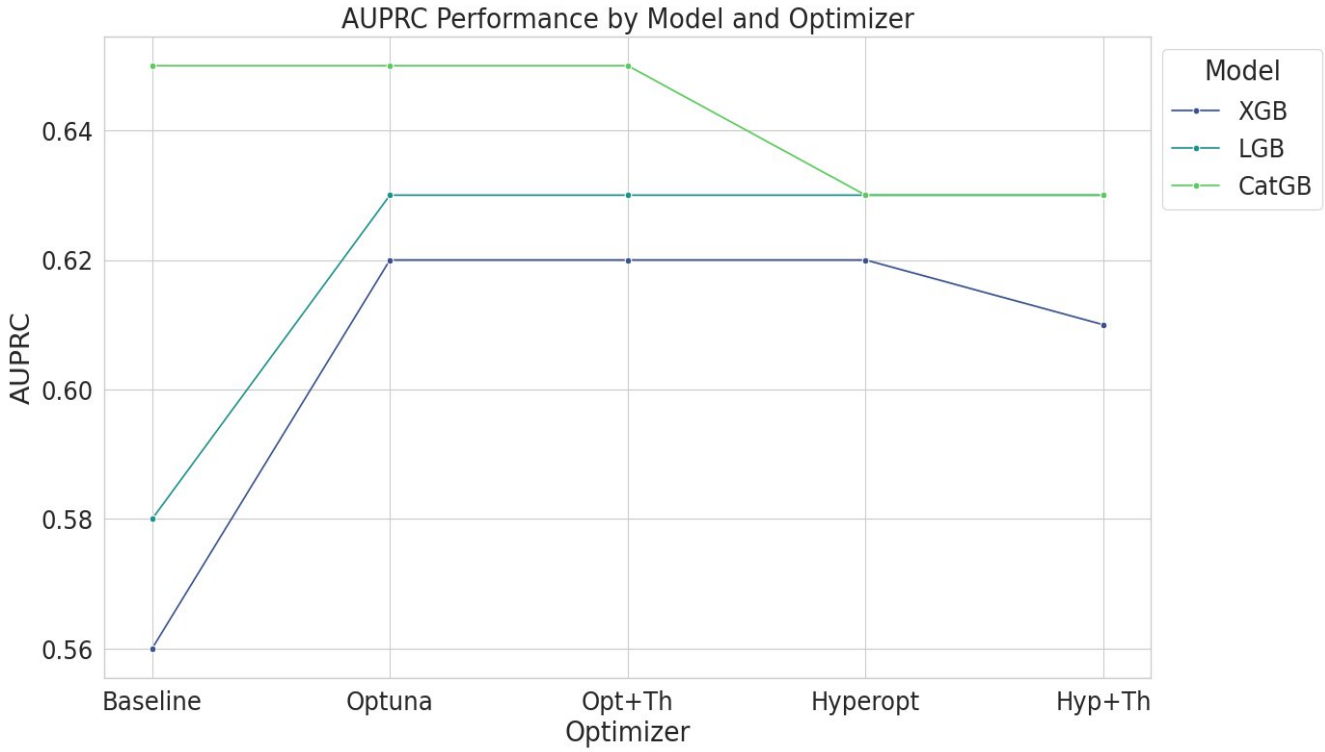
Key Observations:

1. **XGBoost (XGB)**
   - The AUROC scores are consistently high across different optimizers, ranging from 0.78 to 0.83.
   - The optuna, Opt+Th, and Hyp+Th optimizers yield the highest AUROC scores, around 0.83.

2. **LightGBM (LGB)**
   - LightGBM also shows high AUROC scores, with values ranging from 0.79 to 0.83.
   - The Optuna, Opt+Th, and Hyp+Th optimizers improve the AUROC scores to 0.83.

3. **CatBoost (CatGB)**
   - CatBoost has the highest AUROC scores among the three models, starting from 0.83 with the baseline model.
   - The optuna, Opt+Th, and Hyp+Th optimizers maintain the AUROC score at 0.83.

AUPRC (Area Under the Precision-Recall Curve)



The line plot shows the AUPRC variation across different optimization methods for the three gradient boosting models.

Key Observations:

1. **XGBoost (XGB)**
   - The AUPRC scores range from 0.56 to 0.62.
   - The Optuna and Opt+Th optimizers improve the AUPRC scores to 0.62.

2. **LightGBM (LGB)**
   - The AUPRC scores range from 0.58 to 0.62.
   - The highest AUPRC score is achieved with the Optuna and Opt+Th optimizers.

3. **CatBoost (CatGB)**
   - CatBoost shows the highest AUPRC scores, starting from 0.65 with the baseline model.
   - The optuna and Opt+Th optimizers maintain the AUPRC score at 0.65.

Conclusions

Optimizations generally improve model performance across precision, recall, and F1-score metrics. CatBoost shows strong performance even with baseline parameters, but all models benefit significantly from hyperparameter and threshold optimization.

All models demonstrate high AUROC scores, with CatBoost generally outperforming the other models. Optimizations using Optuna, Opt+Th, and Hyp+Th consistently yield the highest AUROC scores.

CatBoost also achieves the highest AUPRC scores, indicating superior performance in precision-recall trade-offs. While optimization techniques improve the AUPRC scores for XGBoost and LightGBM, CatBoost remains the top performer.

Both XGBoost and LightGBM benefit significantly from optimization, showing the largest improvements between baseline and optimized models. In scenarios where recall is the most critical metric, such as medical diagnosis or imbalanced datasets with a minority positive class, XGBoost and LightGBM with optimized hyperparameters and decision thresholds appear to be the best-performing models. However, further testing is recommended to confirm this conclusion.