Purchase Pattern Analysis

Hyperparameter Tuning

December 06, 2024

Overview

This report outlines the hyperparameter tuning process for the LightGBM model, selected as the best-performing model in a supervised machine learning project to predict whether products will be reordered. The objective of tuning was to enhance the model's performance metrics, particularly the F1 score, precision, and recall, while addressing gaps identified during initial evaluation.

Key tuning strategies included optimizing hyperparameters such as learning rate, number of leaves, and maximum depth. The tuned LightGBM model demonstrated an improved balance between precision and recall, solidifying its role as the final model for deployment.

Goals

The goal of this part of the project is to optimize the hyperparameters of the model selected in the previous section - 'LightGBM'. The objective is to improve the performance metrics, mainly F1-score, to make a more accurate prediction of the customer's next purchase.

Model Details

LightGBM is an open-source, high-performance implementation of a gradient boosting framework developed by Microsoft. It is optimized for both speed and memory usage, and it excels for large datasets

Why LightGBM for This Project?

1. Efficiency: LightGBM's ability to handle large datasets and high-dimensional data made it ideal for analyzing purchase history and customer behavior.
2. Performance: Its leaf-wise growth strategy resulted in higher F1 scores compared to other models.
3. Scalability: LightGBM's speed and support for distributed computing ensure scalability for real-world applications.

Model Details - LightGBM

LightGBM is an open-source, high-performance implementation of a gradient boosting framework developed by Microsoft. It is optimized for both speed and memory usage, and it excels for large datasets

Why LightGBM for This Project?

1. Efficiency: LightGBM's ability to handle large datasets and high-dimensional data made it ideal for analyzing purchase history and customer behavior.
2. Performance: Its leaf-wise growth strategy resulted in higher F1 scores compared to other models.
3. Scalability: LightGBM's speed and support for distributed computing ensure scalability for real-world applications.

Hyperparameter Tuning Methodology

Hyperparameters Selected

Learning Parameters:

- n_estimators - Number of boosting rounds
- num_leaves - Maximum number of leaves in a tree. Larger values increase accuracy but risk overfitting
- max_depth - Maximum Depth of a tree

- min_data_in_leaf - Minimum number of samples per leaf. Higher Values prevent overfitting

Regularization Parameters:

- lambda_l1 - L1 regularization term on weights. Adds sparsity to the model
- lambda_l2 - L2 regularization term on weights. Helps control overfitting
- min_gain_to_split - Minimum loss reduction required to split a leaf node. Larger values make the model training faster

Data Sampling Parameters:

- bagging_fraction - Fraction of data used for each iteration (for bagging). Helps with overfitting
- bagging_freq - Frequency of bagging. Also helps deal with overfitting

Tuning Technique - Bayesian Optimization

Bayesian optimization is an advanced and efficient technique used for hyperparameter tuning in machine learning models. Unlike traditional methods like grid search or random search, which involve exploring hyperparameter space blindly, Bayesian optimization uses a probabilistic approach to intelligently navigate and find the best set of hyperparameters. This is particularly useful when tuning complex models like LightGBM or XGBoost, where the hyperparameter space is large and non-linear.

Data Splitting Strategy

As done in the previous section, only 30% sample of the preprocessed data using stratified sampling, has been used for hyperparameter tuning using train_test_split at 80-20 split for train and test set respectively.

Metric Used - F1 Score

The F1 score is the harmonic mean of precision and recall. It thus symmetrically represents both precision and recall in one metric. Tuning the hyperparameter using this metric will help preserve the balance between precision and recall without sacrificing either metric.

Experiments and Results

Baseline Performance

The model's performance without any hyperparameter tuning can be seen by observing following metrics

- Accuracy score: 0.8244693150811636
- Precision score: 0.8655902939295267
- Recall score: 0.8310742028201695
- F1 score: 0.8479811593564383

Distribution of Hyperparameters

Hyperparameter	Distribution	Range
n_estimators	Discrete	[50,1000]
num_leaves	Discrete	[20, 200]
max_depth	Discrete	[-1, 15]
min_data_in_leaf	Discrete	[10, 100]
lambda_l1	Log-Uniform	[0.0001, 10.0]
lambda_l2	Log-Uniform	[0.0001, 10.0]
min_gain_to_split	Log-Uniform	[0.0001, 1.0]
bagging_fraction	Uniform	[0.6, 1.0]
bagging_freq	Discrete	[1, 10]

Best Parameters

Parameters	Values
n_estimators	971
num_leaves	190
max_depth	14
min_data_in_leaf	92
lambda_l1	0.3439142698852568
lambda_l2	1.3288318289273937
min_gain_to_split	0.24035195998981623
bagging_fraction	0.699417894976562
bagging_freq	9

Performance Gains

Below is a comparison of the Base Model (before tuning) and the Tuned Model (after tuning):

Metric	Base Model	Tuned Model	Improvement
Accuracy	82.44%	82.59%	+0.15%
Precision	86.56%	86.76%	+0.20%
Recall	83.10%	83.13%	+0.03%
F1 score	84.80%	84.91%	+0.11%

Interpretation

- Accuracy - The improvement suggests that tuned model correctly classifies 0.15% of data more than that of base model
- Precision - The improvement suggests that the number of false positives has reduced in the tuned model.

- Recall - The small improvement suggests that the tuned model captures slightly more positive cases.
- F1 Score - The improvement of 0.11% suggests that the tuned model balances the classification of both positive and negative cases better than the base model.

Even though tuning resulted in small gains, they may lead to an improvement in making significant business decisions. Even an increase of 0.15% in accuracy may lead to making thousands of more accurate predictions.

Consistency - The model has shown improvement in all the metrics, instead of focusing on one aspect over the other, which signifies a robust performance.

Conclusion

The hyperparameter tuning process successfully enhanced the model's predictive performance. While the absolute gains in metrics like accuracy and F1 score appear modest, their significance in large-scale, real-world scenarios cannot be overstated. The improved model is better equipped to make precise and balanced predictions, ensuring its practical applicability for optimizing marketing strategies and improving customer experiences.