The Chinese University of Hong Kong, Shenzhen

MAT3300 Final Project Report

Dynamic Assets Allocation Based on Different Style Machine Learning Models Using Mean-Variance or Black-Litterman?

Author: Kangqi Yu *Student ID:* 121090735

DECEMBER 2023

Contents

1 Introduction

1.1 Motivation

Modern Portfolio Theory (MPT), as proposed by Markowits[9] (1952), stands as a pivotal cornerstone within modern financial portfolio theory. Markowitz's proposition advocates for the consideration of both risk and expected return in the allocation of assets. Nevertheless, when put into practical application, the resultant portfolio encounters diverse issues, including the emergence of extreme weights[7] (Green & Hollifield, 1992), large sensitivity of the solution to the input expected returns[11] (Best & Grauer, 1991). So, in order to address these issues, Black & Litterman[1] (1991) proposed the Black-Litterman model, which can take investors views into account and produce a posterior distribution of expected returns. But, in the age of machine learning, we already have had the capacity to train some models with prediction power. So, I wonder if we have a fairly great prediction model, can we use it directly to MPT model, or it is better to use Black-Litterman to combine the piror infomation of market?

1.2 Literature Review

Machine learning has been an integral part of the financial services sector for more than four decades. Its recent surge in adoption has significantly impacted investment management and trading practices. Machine learning presents a broader framework for financial modeling compared to its linear parametric predecessors, extending conventional modeling approaches such as factor modeling, derivative pricing, portfolio optimization, and hedging through model-free and data-driven methodologies[3] (Dixon & Halperin, 2019).

In the domain of ML applied to finance, a spectrum of modeling techniques has emerged, prominently featuring predictive modeling and financial modeling. Various methods, including time series analysis, shrinkage estimation, support vector machine, diverse tree-based algorithms, and deep learning models, have gained prominence. The challenge lies in discerning the optimal model and technique as different approaches yield varying outcomes.

For the combination between machine learning model and portfolio optimization, it is not an old topic, Ma, etc.[13] (2021) proposed that they construct a model combine MPT and random forest (RF) work well in Chinese daily investment. Min, etc.[10] (2021) show the combination of Black-Litterman and Random Forest can beat the market index, equal weighted portfolio and worst-c portfolio; Pavan[4] (2018) shows the combination of Black-Litterman and machine learning models can overperform the 1/N portfolio, but sometime the weights of portfolio are so concentrated on some assets. Everyone is arguing they find a powerful model combine MPT or Black-Litterman with machine learning models, but no one have an clear idea that when should we use MPT and when should we use Black-Litterman with a machine learning model combined.

1.3 My Contribution

To find out the answer to the above problem, I trianed two different style machine learning models, and use they to combine with MPT and Black-Litterman model respectively. Then, I compare the performance of these models and try to find out the empirical best model to use in different situation. The final conclusions is the model consider more about risk is more suitable to use Black-Litterman to do optimization, and the model that is aggressive is more suitable to use MPT to do the portfolio optimization. If you need to take transaction into account, Black-Litterman has lower turnover ratio definitely. The choice of you may change.

2 Data

2.1 Data Source

The data I used in this paper is collected from Wind and CSMAR with the range from the beginning of 2016 to the end of 2022. In details, the return data is constructed manually through the close price data from Wind; the three momentum factors are constructed manually from the close price data and the related return data from Wind; the periodical/quarter factors are constructed through data from the last period from Wind and CSMAR.

2.2 Stock Selection

To reduce the number of correlation coefficients I need to estimate and the protential estimation errors, I choose ten stocks from the automotive industry: 000338.SZ WEICHAI POWER, 000625.SH CHANGAN AUTOMOBILE, 000951.SZ CNHTC-JNTC, 002594.SZ BYD, 600104.SH SAIC MOTOR, 600418.SH JAC, 600660.SH FYG, 601238.SH GAC GROUP, 601633.SH GREAT WALL MOTOR, 601689.SH TUOPU GROUP. Such selection may potentially yield better factor performance, offer comparability, and eliminate the need for industry neutralization.

2.3 Factor Filtration

For factor filtration, I computed the correlation between the return and the factors, and chose factors with $IC > 0.01$, $IR > 0.03$ with $IC_{At} = Corr(f_{At}, r_t)$, f_{At}: Factor A loading at time t, r_t: Stock return at time $t+1$, $IC_A = Mean(IC_{At})$, $IR_A = Mean(IC_{At})/std(IC_{At})$. After filtration, I get 27 factors including 6 valuation factors like Dividend Yields, 3 momentum factors like 20-Day Momentum, 6 risk analysis factors like 30-Day Variance and 12 financial analysis factors like Cash Recovery Ratio and Cost of Goods Sold Ratio. (Details about factors can be seen from the table in the appendix.)

2.4 Data Processing

There are some missing value in the return column, I choose to use 0 to fill the NAs in returns for the mean of returns is almost 0 and the most occasion that missing return means the stock is suspended and the price of it should have no changes.

2.5 Data Splitting

I take the data in the range from 2016-01-01 to 2021-12-31 as the insample training data and the data in the range from 2022-01-01 to 2022-12-31 as the outsample testing data.

3 Model Construction and Analysis

3.1 Overview

In this paper, based on my previous research experience, I design two tree models to predict the expected return and generate views. Then, I will use MPT and Black-Litterman approach to get the optimal weight on these ten stocks.

3.2 Machine Learning models - Trees

3.2.1 Traditional Advanced Tree Models

Boosting[5] (Friedman, 2001) and bagging[2] (Breiman, 1996), prominent ensemble learning techniques, construct models differently. Bagging creates diverse base decision trees from random subsets of the training data, aggregating their predictions to reduce overfitting. Contrastingly, Boosting iteratively builds a series of weak classifiers, focusing on correcting predecessor errors.

Both methods significantly enhance model performance. Bagging, like Random Forests, handles large data and features effectively, reducing variance for better stability. Boosting, including XGBoost and lightGBM, emphasizes misclassified samples, enhancing predictions.

3.2.2 My Tree Models

In this paper, I would like to introduce a new tree model combined with boosting and bagging method.

Feature Engineering Given that nearly half of my factors are tied to quarterly data or periodical data, resulting in approximately 60 trading days with identical factor values, I have implemented a novel feature to discern these instances from one another. Referred to as *continuous counting*, this feature entails recording the number of days following the publication date. Its primary utility lies in capturing the influence of these quarterly data points on daily stock returns, accounting for the passage of time. This innovative addition enables us to better gauge and incorporate the temporal impact of these periodic data fluctuations on my predictive models for stock returns.

Besides, I assign more weights on the new data and less weights on the old data to learn the new trend in the market. The weight of the data point is calculated by the following formula:

$$w_{item_t} = \frac{rank(item_t)}{\sum_{i=1}^{T} rank(item_i)}$$

.

Ensemble Learning I use bagging on two type of boosting models, lightGBM and XGBoost, rather than just use one to predict to reduce the risk to be overfitting. The following Figure 1 show my idea exactly.

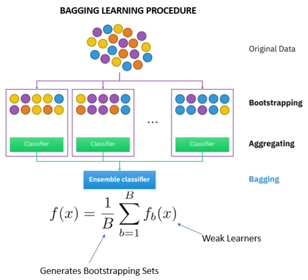


Figure 1: my model architecture

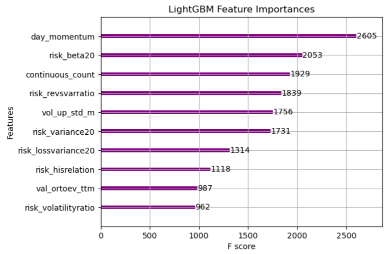


Figure 2: Feature importance of lightGBM

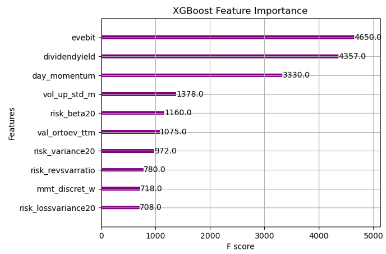


Figure 3: Feature importance of XGBoost

3.2.3 Feature Importance

After training the data, I get the importance (the split number of each feature) of each features, which is shown in the above figure. As we can see, lightGBM forest capture the features brought by *continuous counting* and all the risk factors, while XGBoost forest capture the features brought by *20day_momentum*, *dividendyield*, and *evebit*, which are more aggressive factors. So, they are two models with different style, just like what we are searching.

4 Morden Portfolio Theory (MPT)

Morden Portfolio Theory utilize a mean variance framework to construct an optimal portfolio. The key steps involved in the MPT include:

a. Estimating the expected returns of assets. In this paper, I use the prediction of XGBoost forest or lightGBM forest.

b. Estimating the covariance matrix of asset returns. In this paper, I use the exponential dacaying method with λ as 0.94 to estimate the covariance matrix of asset returns. The decaying formula is as follows:

$$w_t = \frac{\lambda^{T-t+1}}{\sum_{t=1}^{T} \lambda^t}$$

c. Calculating the optimal weights of assets. This is typically done using the formula:

$$\begin{aligned} \text{minimize} \quad & \mathbf{w}^\top \mathbf{R} - \frac{\gamma}{2}\mathbf{w}^\top \Sigma \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{1} = 1, \quad \mathbf{w} \geq \mathbf{0} \end{aligned}$$

where $\mathbf{w}$ is the weight we need to optimize, $\mathbf{R}$ is the expected return vector, and γ is the risk aversion.

5 Black-Litterman Model (BL)

The Black-Litterman model is a method used to combine the views of an investor with market equilibrium to form an expected return on assets. It addresses the limitations of the traditional mean-variance optimization by incorporating subjective views into the asset allocation process. Now, I use lightGBM forest or XGBoost forest to generate investment views dynamically.

The key steps involved in the Black-Litterman model include:

a. The prior expected return is generated from the solution of mean variance framework:

$$x = \frac{1}{\gamma}\Sigma^{-1}\mu$$

Sharpe[12] (2007) proposed that the market proposition (w) is the optimal assets weights, so the prior expected return is

$$\mu_0 = \gamma \Sigma w$$

b. Estimating the covariance matrix of asset returns using exponential dacaying with λ as 0.94[8] (J.P. Morgan, 1996). The decaying formula is as follows:

$$w_t = \frac{\lambda^{T-t+1}}{\sum_{t=1}^{T} \lambda^t}$$

c. Incorporating investor views or opinions into the model. These views can be expressed as a identity matrix and a return predicted vector by XGBoost forest.

d. Estimate the uncertainty of the views by using the formula (He & Litterman's method[6] 2002):

$$\Omega = diag(\tau P \Sigma P^{\top})$$

where Ω is the covariance matrix of the views, $diag()$ is a function to preserve only the diagonal elements in the matrix, τ is the scaling factor, and in this paper, it is 0.05, Σ is the covariance matrix of asset returns, P is the matrix that translates views into returns.

e. Calculating the implied returns by combining the prior expected return and the views from lightGBM forest or XGBoost forest. This is typically done using the formula:

$$\mu_p = \left((\tau\Sigma)^{-1} + P^{\top}\Omega^{-1}P\right)^{-1}\left((\tau\Sigma)^{-1}\mu_0 + P^{\top}\Omega^{-1}q\right)$$

where μ_p is the posterior expected return vector, μ_0 is the prior expected return vector and Q is the vector of views.

f. Updating the portfolio allocation based on mean variance framework (just like the Morden Portfolio Theory section).

The Black-Litterman model provides a framework for investors to incorporate their subjective views into the portfolio optimization process, resulting in a more customized asset allocation strategy.

6 Model Validation

6.1 Tree Models Performance

I use mse, IC and IR to evaluate the performance of my models. The following table shows the result of my models. From the following table, we can see the insample performance of lightGBM is exaggerated compared to the insample performance of XGBoost, benifiting from the leaf-wise growing strategy. The performance of lightGBM and XGBoost is almost the same based on outsample IC and IR, and they are higher than all factors in the appendix.

Model and dataset	Metrics			
	mse	R^2	IC	IR
LightGBM Insample	0.00010	0.90170	0.82614	4.80196
LightGBM Outsample	0.00091	0.00010	0.04098	0.10894
XGBoost Insample	0.00046	0.68502	0.59617	1.73243
XGBoost Outsample	0.00079	0.00207	0.03940	0.10844

Table 1: Tree models performance

6.2 MPT and BL Performance

6.2.1 Metrics

I choose three exceeded metrics to reflect the performance. All of them are relative metrics. In this paper, I use the models used MPT as baselines model.

Exceeded Return It is just the return of BL model higher than MPT model at the end of outsample period.

$$ER = R_{BL,T} - R_{MPT,T}$$

Exceeded Sharpe It is the Sharpe value of daily exceeded return taking MPT model as baselines.

$$ES = \frac{mean(R_{BL,t} - R_{MPT,t})}{std(R_{BL,t} - R_{MPT,t})}$$

Exceeded Average Turnover Here is the formula to calculate turnover based on the weights:

$$Turnover_t = \sum |\Delta weights_t|$$

The formula to calculate the exceeded average turnover is:

$$EAT = mean(Turnover_{LB,t}) - mean(Turnover_{MPT,t})$$

6.2.2 Method

I conduct a sensitivity analysis in this section. I set risk aversion as 0.1, 0.5, 1, 2, 5, 10 and do the outsample test respectively. From the following figures we can see, when risk aversion changing, the exceeded performance of BL to MPT is changing too. But for most of cases, the sign of each performance metrics is not changed, which means there truly have some situation we should use BL to do the portfolio optimization or use MPT to do the portfolio optimization.

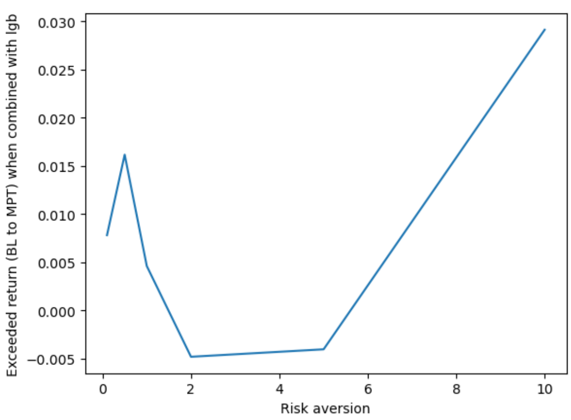


Figure 4: The exceeded return (BL to MPT) when combined with lgb

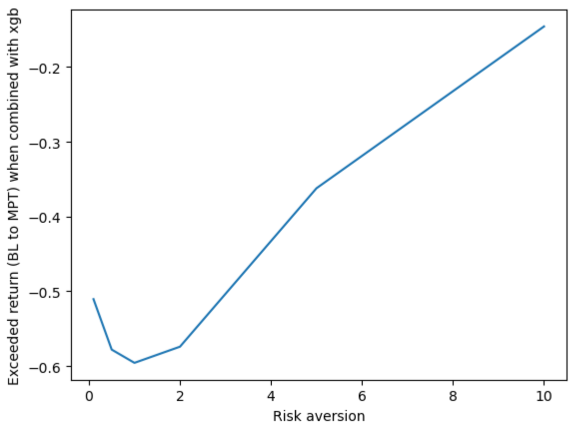


Figure 5: The exceeded return (BL to MPT) when combined with xgb

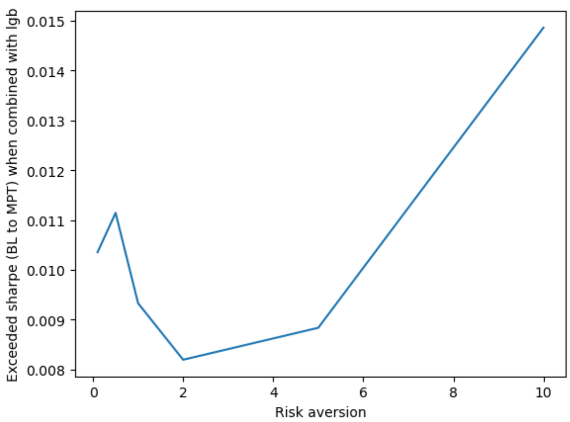


Figure 6: The exceeded sharpe (BL to MPT) when combined with lgb

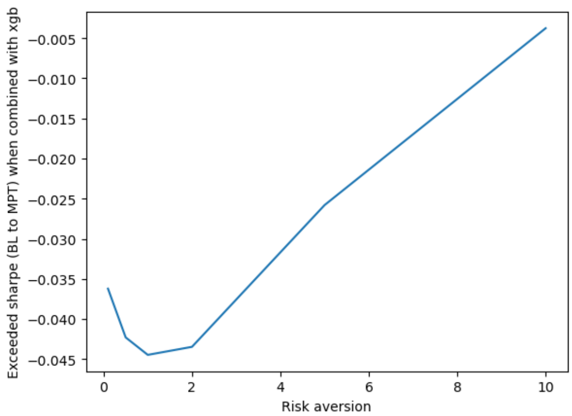


Figure 7: The exceeded sharpe (BL to MPT) when combined with lgb

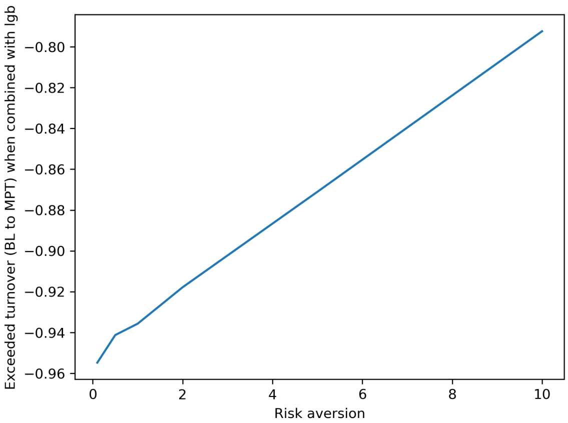


Figure 8: The exceeded turnover (BL to MPT) when combined with lgb

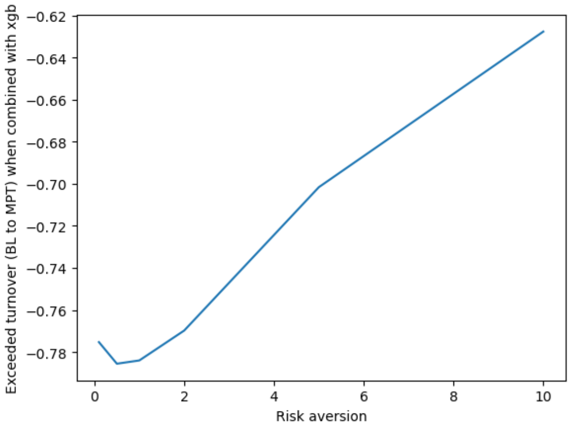


Figure 9: The exceeded turnover (BL to MPT) when combined with lgb

7 Conclusions

From the empirical result above, we can see Black-Litterman always has lower turnover ratio than MPT, ethier combined with lightGBM or XGBoost. which means if we choose Black-Litterman, we will save a lot of transaction cost. If we do not take transaction cost into account, Black-Litterman is a suitable model to optimize with the machine learning model focusing more on the market risk and MPT is a suitable model to optimize with the machine learning more aggressive.

References

[1] Black Fischer & Litterman Robert B. Asset allocation, 1991.

[2] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[3] Igor Dixon, Matthew & Halperin. The four horsemen of machine learning in finance. *SSRN Electronic Journal*, 01 2019.

[4] Pavan Donthireddy. Black-litterman portfolios with machine learning derived views. 07 2018.

[5] Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[6] Guangliang He and Robert B. Litterman. The intuition behind black-litterman model portfolios. *Capital Markets: Asset Pricing & Valuation*, 2002.

[7] Richard C. Green & Burton Hollifield. When will mean-variance efficient portfolios be well diversified? *The Journal of Finance*, 47(5):1785–1809, 1992.

[8] Morgan J.P. Riskmetrics technical document. 1996.

[9] Markowits & Harry M. Portfolio selection. *Journal of finance*, 7(1):71–91, 1952.

[10] Liangyu Min, Jiawei Dong, Dewen Liu, and Xiangxi Kong. A black-litterman portfolio selection model with investor opinions generating from machine learning algorithms. *Engineering Letters*, 29:710–721, 05 2021.

[11] Best Michael J & Grauer Robert R. On the sensitivity of mean-variance-efficient portfolios to changes in asset means: some analytical and computational results. *The review of financial studies*, 4(2):315–342, 1991.

[12] William F. Sharpe. Expected utility asset allocation. *Financial Analysts Journal*, 63(5):18–30, 2007.

[13] Yilin Ma & Ruizhu Han & Weizhong Wang. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165:113973, 2021.

A Factors

Factor	Classification	Frequency	Description
Dividend Yield(%)	Valuation Factor	Daily	Total annual dividend payments/ Stock market capitalization
val_lnmv	Valuation Factor	Daily	Logarithmic market capitalization
val_lntotassets	Valuation Factor	Periodically	Logarithmic total assets
val_ortoev_ttm	Valuation Factor	Daily	Operating Income (TTM)/Enterprise Value
dividendyield2	Valuation Factor	Daily	Dividend Yield (12 Months)
val_floatmv	Valuation Factor	Daily	Floating Market Cap_PIT
20day_momentum	Momentum Factor	Daily	20-Day Momentum = Closing Price of the day / (Average of the previous 20 days' closing prices)
mmt_discret_W	Momentum Factor	Daily	One-week information dispersion momentum: the difference between the number of days with positive returns and the number of days with negative returns in the past five trading days.
Vol_up_std_M	Momentum Factor	Daily	One-month upward volatility: the standard deviation of positive returns over the first twenty trading days
risk_variance20	Risk Analysis	Daily	20-Day Variance_PIT
risk_lossvariance20	Risk Analysis	Daily	20-Day Loss Variance_PIT
risk_beta20	Risk Analysis	Daily	20-Day Beta_PIT

Factor	Classification	Frequency	Description
risk_volatilityratio	Risk Analysis	Daily	Ratio of Individual Security Volatility and Market Volatility_PIT
risk_hisrelation	Risk Analysis	Daily	252-Day Correlation between Individual Security and the Market_PIT
risk_revsvarratio	Risk Analysis	Daily	30-Day Variance / 120-Day Variance_PIT
fa_cashrecovratio_ttm	Financial Analysis	Periodically	Cash Recovery Ratio (TTM)_PIT
fa_blev	Financial Analysis	Periodically	Book Leverage_PIT
fa_current	Financial Analysis	Periodically	Current Ratio_PIT
fa_apturn_ttm	Financial Analysis	Periodically	Accounts Payable Turnover (TTM)_PIT
fa_ncgr_ttm	Financial Analysis	Periodically	Growth Rate - Net Cash Flow (TTM)_PIT
fa_salestocost_ttm	Financial Analysis	Periodically	Cost of goods sold ratio (TTM)
fa_sellexpensetogr_ttm	Financial Analysis	Periodically	Selling expenses/total operating income (TTM)
enebit	Financial Analysis	Quarterly	Enterprise Value multiplier: Total market value/EBITDA
fa_operincometopbt	Financial Analysis	Periodically	Net Income from Operating Activities / Total Profit_PIT
fa_octogr_ttm	Financial Analysis	Periodically	Total Operating Cost / Total Operating Revenue (TTM)_PIT
fa_netprofitmargin_ttm	Financial Analysis	Periodically	Net Profit Margin (TTM)_PIT

Factor	Classification	Frequency	Description
fa_salescashtoor	Financial Analysis	Periodically	Cash Received from Sales of Goods and Rendering of Services / Operating Revenue_PIT

B Codes

Listing 1: Factor Selection

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def cal_ICIR(data: pd.DataFrame, feild: str) -> tuple[float,
    float]:
    """
    data is a dataframe with columns: date, return, factor feild
    feild is the factor name
    return IC and IR
    """
    data = data.loc[data.loc[:, 'date'] < "2022-01-01", ['date',
        'return', feild]]
    data.dropna(inplace=True)
    IC_dataframe = data.groupby('date').apply(lambda x: x.corr(
        method='spearman')[feild]['return'])
    return IC_dataframe.mean(), IC_dataframe.mean()/IC_dataframe
        .std()

def test_factor(ICIR: tuple[float, float]) -> str:
    """
    ICIR is a tuple of IC and IR
    return the test result
    """
    if abs(ICIR[0]) > 0.01 and abs(ICIR[1]) > 0.03:
        return 'pass'
    else:
        return 'fail'
```

```
if __name__ == '__main__':
    data = pd.read_csv('factors.csv', index_col=0)
    colname = data.columns.tolist()
    other_info = colname[:3]
    factorname = colname[3:]
    pass_list = []
    for i in factorname:
        if i == 'return':
            continue
        ICIR = cal_ICIR(data, i)
        print(i)
        print(ICIR)
        result = test_factor(ICIR)
        if result == 'pass':
            pass_list.append(i)
        print(result)
    data.loc[:, other_info + pass_list + ['return']].to_csv('
      factor_pass.csv')
```

Tree_model

December 17, 2023

[ ]:
```
import optuna
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import KFold
import lightgbm as lgb
import xgboost as xgb
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('factor_pass.csv', index_col=0)
data['continuous_count'] = data.groupby((data['evebit'] != data['evebit'].
 ↪shift(1)).cumsum()).cumcount() + 1
data = data.reindex(columns=data.columns.tolist()[:-2] + ['continuous_count',␣
 ↪'return'])
data.dropna(inplace=True)

insample_data = data.loc[data.loc[:, 'date'] < "2022-01-01", :]
insample_data_sorted = insample_data.sort_values('date')
insample_data_sorted['rank'] = insample_data_sorted['date'].rank()
sum_of_weight = (insample_data_sorted['rank']).sum()
insample_data_sorted['weight'] = insample_data_sorted['rank'] / sum_of_weight
outdsample_data = data.loc[data.loc[:, 'date'] >= "2022-01-01", :]

X = insample_data_sorted.iloc[:, 3:-3].astype(float)
y = insample_data_sorted.iloc[:, -3].astype(float)
other_info_outsample_test = outdsample_data.iloc[:, :3]
X_outsample_test = outdsample_data.iloc[:, 3:-1].astype(float)
y_outsample_test = outdsample_data.iloc[:, -1].astype(float)

def objective_lgb(trial):
    params = {
        'objective': 'regression',
        'metric': 'rmse',
        'boosting_type': 'gbdt',
        'verbosity': -1,
        'lambda_l1': trial.suggest_float('lambda_l1', 1e-8, 10.0, log = True),
        'lambda_l2': trial.suggest_float('lambda_l2', 1e-8, 10.0, log = True),
```

```
        'num_leaves': trial.suggest_int('num_leaves', 2, 256),
        'feature_fraction': trial.suggest_float('feature_fraction', 0.2, 1.0),
        'bagging_fraction': trial.suggest_float('bagging_fraction', 0.2, 1.0),
        'bagging_freq': trial.suggest_int('bagging_freq', 0, 8),
        'min_child_samples': trial.suggest_int('min_child_samples', 5, 100),
        'max_depth': trial.suggest_int('max_depth', 2, 30),
        'random_state': 42
    }
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    avg_rmse = 0.0

    for train_idx, val_idx in kf.split(X):
        X_train, X_val = X.iloc[train_idx], X.iloc[val_idx]
        y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]
        weight_train, weight_val = insample_data_sorted.iloc[train_idx, -1],␣
↪insample_data_sorted.iloc[val_idx, -1]

        lgb_train = lgb.Dataset(X_train, label=y_train, weight=weight_train)
        lgb_val = lgb.Dataset(X_val, label=y_val, weight=weight_val)

        model = lgb.train(
            params,
            lgb_train,
            valid_sets=[lgb_train, lgb_val]
        )

        preds = model.predict(X_val, num_iteration=model.best_iteration)
        rmse = mean_squared_error(y_val, preds, squared=False)
        avg_rmse += rmse / 5  # Average RMSE over folds

    return avg_rmse

def objective_xgb(trial):
    params = {
        'objective': 'reg:squarederror',
        'verbosity': 0,
        'lambda': trial.suggest_float('lambda', 1e-8, 10.0, log = True),
        'alpha': trial.suggest_float('alpha', 1e-8, 10.0, log = True),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.2, 1.0),
        'subsample': trial.suggest_float('subsample', 0.2, 1.0),
        'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.2),
        'n_estimators': trial.suggest_int('n_estimators', 100, 1000),
        'max_depth': trial.suggest_int('max_depth', 2, 30),
        'random_state': 42,
        'baggling_fraction': trial.suggest_float('bagging_fraction', 0.2, 1.0),
        'feature_fraction': trial.suggest_float('feature_fraction', 0.2, 1.0),
        'bagging_freq': trial.suggest_int('bagging_freq', 0, 8),
```

```
        'min_child_samples': trial.suggest_int('min_child_samples', 5, 100),
    }
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    avg_rmse = 0.0

    for train_idx, val_idx in kf.split(X):
        X_train, X_val = X.iloc[train_idx], X.iloc[val_idx]
        y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]
        weight_train, weight_val = insample_data_sorted.iloc[train_idx, -2],␣
↪insample_data_sorted.iloc[val_idx, -2]

        dtrain = xgb.DMatrix(X_train, label=y_train, weight = weight_train)
        dval = xgb.DMatrix(X_val, label=y_val, weight = weight_val)

        model = xgb.train(params, dtrain, evals=[(dtrain, 'train'), (dval,␣
↪'eval')], verbose_eval=False, early_stopping_rounds=50)
        preds = model.predict(dval)
        rmse = mean_squared_error(y_val, preds, squared=False)
        avg_rmse += rmse / 5  # Taking average RMSE over folds

    return avg_rmse
```

[ ]:
```
study_lgb = optuna.create_study(direction='minimize', sampler=optuna.samplers.
↪TPESampler(seed=42))
study_lgb.optimize(objective_lgb, n_trials=100)

best_params_lgb = study_lgb.best_params
best_rmse_lgb = study_lgb.best_value

study_xgb = optuna.create_study(direction='minimize', sampler=optuna.samplers.
↪TPESampler(seed=42))
study_xgb.optimize(objective_xgb, n_trials=100)

best_params_xgb = study_xgb.best_params
best_rmse_xgb = study_xgb.best_value
```

```
[I 2023-12-13 10:13:31,206] A new study created in memory with name: no-name-
dde4da1b-4f7a-4a63-8a90-52a09d270d65
[I 2023-12-13 10:13:32,914] Trial 0 finished with value: 0.025383145516882366
and parameters: {'lambda_l1': 2.348881295853308e-05, 'lambda_l2':
3.6010467344475403, 'num_leaves': 188, 'feature_fraction': 0.6789267873576292,
'bagging_fraction': 0.3248149123539492, 'bagging_freq': 1, 'min_child_samples':
10, 'max_depth': 27}. Best is trial 0 with value: 0.025383145516882366.
[I 2023-12-13 10:13:33,376] Trial 1 finished with value: 0.025400561065519283
and parameters: {'lambda_l1': 0.002570603566117598, 'lambda_l2':
0.023585940584142682, 'num_leaves': 7, 'feature_fraction': 0.9759278817295955,
'bagging_fraction': 0.8659541126403374, 'bagging_freq': 1, 'min_child_samples':
```

```
22, 'max_depth': 7}. Best is trial 0 with value: 0.025383145516882366.
[I 2023-12-13 10:13:35,265] Trial 2 finished with value: 0.025770321119838478
and parameters: {'lambda_l1': 5.472429642032198e-06, 'lambda_l2':
0.00052821153945323, 'num_leaves': 112, 'feature_fraction': 0.43298331215843355,
'bagging_fraction': 0.6894823157779035, 'bagging_freq': 1, 'min_child_samples':
33, 'max_depth': 12}. Best is trial 0 with value: 0.025383145516882366.
[I 2023-12-13 10:13:36,025] Trial 3 finished with value: 0.025330062266094012
and parameters: {'lambda_l1': 0.00012724181576752517, 'lambda_l2':
0.1165691561324743, 'num_leaves': 52, 'feature_fraction': 0.6113875507308892,
'bagging_fraction': 0.6739316550896339, 'bagging_freq': 0, 'min_child_samples':
63, 'max_depth': 6}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:37,283] Trial 4 finished with value: 0.025354971750557925
and parameters: {'lambda_l1': 3.850031979199519e-08, 'lambda_l2':
3.4671276804481113, 'num_leaves': 248, 'feature_fraction': 0.846717878493169,
'bagging_fraction': 0.4436910153386966, 'bagging_freq': 0, 'min_child_samples':
70, 'max_depth': 14}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:37,842] Trial 5 finished with value: 0.025573046739679514
and parameters: {'lambda_l1': 1.254134495897175e-07, 'lambda_l2':
0.00028614897264046574, 'num_leaves': 10, 'feature_fraction':
0.9274563216630256, 'bagging_fraction': 0.40702398528001354, 'bagging_freq': 5,
'min_child_samples': 34, 'max_depth': 17}. Best is trial 3 with value:
0.025330062266094012.
[I 2023-12-13 10:13:38,330] Trial 6 finished with value: 0.02540053423583984 and
parameters: {'lambda_l1': 0.0008325158565947976, 'lambda_l2':
4.609885087947832e-07, 'num_leaves': 249, 'feature_fraction':
0.8201062586888916, 'bagging_fraction': 0.9515991532513512, 'bagging_freq': 8,
'min_child_samples': 62, 'max_depth': 28}. Best is trial 3 with value:
0.025330062266094012.
[I 2023-12-13 10:13:38,959] Trial 7 finished with value: 0.02556629922247531 and
parameters: {'lambda_l1': 6.257956190096665e-08, 'lambda_l2':
5.805581976088804e-07, 'num_leaves': 13, 'feature_fraction': 0.4602642646106115,
'bagging_fraction': 0.5109418317515857, 'bagging_freq': 2, 'min_child_samples':
84, 'max_depth': 12}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:39,599] Trial 8 finished with value: 0.025955591866288708
and parameters: {'lambda_l1': 3.376063348877853e-06, 'lambda_l2':
0.0007660634613082914, 'num_leaves': 37, 'feature_fraction': 0.8417575846032317,
'bagging_fraction': 0.2596405149438167, 'bagging_freq': 8, 'min_child_samples':
79, 'max_depth': 7}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:40,369] Trial 9 finished with value: 0.02533367332188062 and
parameters: {'lambda_l1': 1.1212412169964432e-08, 'lambda_l2':
0.2183498289760726, 'num_leaves': 182, 'feature_fraction': 0.7832057344327898,
'bagging_fraction': 0.8170162773487566, 'bagging_freq': 0, 'min_child_samples':
39, 'max_depth': 5}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:40,965] Trial 10 finished with value: 0.025400561065519283
and parameters: {'lambda_l1': 0.3084620517909565, 'lambda_l2':
1.1323342574942026e-08, 'num_leaves': 80, 'feature_fraction':
0.21436950049185588, 'bagging_fraction': 0.6452374670527004, 'bagging_freq': 4,
'min_child_samples': 97, 'max_depth': 2}. Best is trial 3 with value:
```

0.025330062266094012.
[I 2023-12-13 10:13:41,482] Trial 11 finished with value: 0.025400561065519283 and parameters: {'lambda_l1': 0.023623375641233485, 'lambda_l2': 0.06112562154158994, 'num_leaves': 168, 'feature_fraction': 0.6788645195964429, 'bagging_fraction': 0.7731769493484759, 'bagging_freq': 3, 'min_child_samples': 49, 'max_depth': 2}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:41,870] Trial 12 finished with value: 0.025400561065519283 and parameters: {'lambda_l1': 1.9404675880825022, 'lambda_l2': 0.2517867400097501, 'num_leaves': 185, 'feature_fraction': 0.6145273325314549, 'bagging_fraction': 0.7748607714713163, 'bagging_freq': 0, 'min_child_samples': 49, 'max_depth': 20}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:42,657] Trial 13 finished with value: 0.025386709226214544 and parameters: {'lambda_l1': 9.28077866199323e-05, 'lambda_l2': 8.128983117583447, 'num_leaves': 68, 'feature_fraction': 0.739091319152402, 'bagging_fraction': 0.5772738979073035, 'bagging_freq': 6, 'min_child_samples': 57, 'max_depth': 8}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:43,754] Trial 14 finished with value: 0.025393644986868413 and parameters: {'lambda_l1': 1.0521055418702655e-06, 'lambda_l2': 0.00934669040346174, 'num_leaves': 129, 'feature_fraction': 0.5691238715895103, 'bagging_fraction': 0.9760769947709625, 'bagging_freq': 3, 'min_child_samples': 43, 'max_depth': 7}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:44,393] Trial 15 finished with value: 0.02533506427191585 and parameters: {'lambda_l1': 6.731180946161928e-05, 'lambda_l2': 0.3996362846812158, 'num_leaves': 167, 'feature_fraction': 0.762950185127336, 'bagging_fraction': 0.7086724951595825, 'bagging_freq': 0, 'min_child_samples': 68, 'max_depth': 4}. Best is trial 3 with value: 0.025330062266094012.
[I 2023-12-13 10:13:47,164] Trial 16 finished with value: 0.0253256533736168 and parameters: {'lambda_l1': 1.255748601576442e-08, 'lambda_l2': 0.38364654650778457, 'num_leaves': 222, 'feature_fraction': 0.9835404734181961, 'bagging_fraction': 0.5799999963515658, 'bagging_freq': 2, 'min_child_samples': 5, 'max_depth': 20}. Best is trial 16 with value: 0.0253256533736168.
[I 2023-12-13 10:13:51,738] Trial 17 finished with value: 0.02590726427018697 and parameters: {'lambda_l1': 6.089192651294439e-07, 'lambda_l2': 0.004305080963629689, 'num_leaves': 216, 'feature_fraction': 0.9574439411308509, 'bagging_fraction': 0.5660034089645037, 'bagging_freq': 2, 'min_child_samples': 6, 'max_depth': 23}. Best is trial 16 with value: 0.0253256533736168.
[I 2023-12-13 10:13:52,673] Trial 18 finished with value: 0.02534629952454493 and parameters: {'lambda_l1': 1.1002794580071929e-08, 'lambda_l2': 0.9569917445659704, 'num_leaves': 130, 'feature_fraction': 0.8949745395144866, 'bagging_fraction': 0.6320760160908753, 'bagging_freq': 2, 'min_child_samples': 100, 'max_depth': 23}. Best is trial 16 with value: 0.0253256533736168.
[I 2023-12-13 10:13:53,142] Trial 19 finished with value: 0.025400561065519283 and parameters: {'lambda_l1': 0.004958678358204063, 'lambda_l2': 0.046057826516286884, 'num_leaves': 87, 'feature_fraction': 0.9630144129354888, 'bagging_fraction': 0.4732022922583826, 'bagging_freq': 3, 'min_child_samples': 24, 'max_depth': 18}. Best is trial 16 with value: 0.0253256533736168.
[I 2023-12-13 10:13:53,859] Trial 20 finished with value: 0.025389953689017142 and parameters: {'lambda_l1': 0.00020462912095823503, 'lambda_l2':

```
8.942708947757339, 'num_leaves': 49, 'feature_fraction': 0.8949881074087542,
'bagging_fraction': 0.5454862287997417, 'bagging_freq': 6, 'min_child_samples':
85, 'max_depth': 21}. Best is trial 16 with value: 0.0253256533736168.
[I 2023-12-13 10:13:56,715] Trial 21 finished with value: 0.02530604193384279
and parameters: {'lambda_l1': 1.3858634221553198e-08, 'lambda_l2':
0.2726996057010207, 'num_leaves': 218, 'feature_fraction': 0.9863339033620951,
'bagging_fraction': 0.7561736020872727, 'bagging_freq': 0, 'min_child_samples':
13, 'max_depth': 10}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:13:58,693] Trial 22 finished with value: 0.025328978048792833
and parameters: {'lambda_l1': 1.8719422840595333e-07, 'lambda_l2':
0.7215756291343561, 'num_leaves': 218, 'feature_fraction': 0.8849994787138371,
'bagging_fraction': 0.6885387176620723, 'bagging_freq': 1, 'min_child_samples':
16, 'max_depth': 10}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:00,809] Trial 23 finished with value: 0.025340195248827177
and parameters: {'lambda_l1': 2.2386032062119328e-07, 'lambda_l2':
1.163819060998307, 'num_leaves': 219, 'feature_fraction': 0.9966563156017068,
'bagging_fraction': 0.601389771732912, 'bagging_freq': 1, 'min_child_samples':
16, 'max_depth': 10}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:03,498] Trial 24 finished with value: 0.025326738570416972
and parameters: {'lambda_l1': 2.0200982585537783e-07, 'lambda_l2':
0.878289590676891, 'num_leaves': 221, 'feature_fraction': 0.997975862784779,
'bagging_fraction': 0.7240937094238586, 'bagging_freq': 2, 'min_child_samples':
15, 'max_depth': 14}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:05,459] Trial 25 finished with value: 0.025341158184914385
and parameters: {'lambda_l1': 2.4746362722921658e-08, 'lambda_l2':
1.430659971367418, 'num_leaves': 235, 'feature_fraction': 0.9942740556900019,
'bagging_fraction': 0.7448669300090887, 'bagging_freq': 4, 'min_child_samples':
25, 'max_depth': 15}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:09,292] Trial 26 finished with value: 0.025326284229444496
and parameters: {'lambda_l1': 5.831109144082721e-08, 'lambda_l2':
0.13688429702090743, 'num_leaves': 206, 'feature_fraction': 0.9159866152714637,
'bagging_fraction': 0.8546143331652052, 'bagging_freq': 2, 'min_child_samples':
6, 'max_depth': 13}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:13,026] Trial 27 finished with value: 0.025555504617538326
and parameters: {'lambda_l1': 1.0759418614004682e-08, 'lambda_l2':
0.01644886167704751, 'num_leaves': 152, 'feature_fraction': 0.9253502805088382,
'bagging_fraction': 0.8633508152309205, 'bagging_freq': 3, 'min_child_samples':
7, 'max_depth': 19}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:17,449] Trial 28 finished with value: 0.025394161901315514
and parameters: {'lambda_l1': 5.541772632032647e-08, 'lambda_l2':
0.05657355454282213, 'num_leaves': 201, 'feature_fraction': 0.9098902526951992,
'bagging_fraction': 0.9093534766172033, 'bagging_freq': 4, 'min_child_samples':
5, 'max_depth': 25}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:19,426] Trial 29 finished with value: 0.02535412847833298
and parameters: {'lambda_l1': 1.658990591801076e-05, 'lambda_l2':
3.353866942666003, 'num_leaves': 197, 'feature_fraction': 0.8531091319075019,
'bagging_fraction': 0.8232262658267391, 'bagging_freq': 1, 'min_child_samples':
13, 'max_depth': 11}. Best is trial 21 with value: 0.02530604193384279.
```

```
[I 2023-12-13 10:14:22,359] Trial 30 finished with value: 0.02533153470138293
and parameters: {'lambda_l1': 1.3720620082312516e-06, 'lambda_l2':
0.14206473233944775, 'num_leaves': 253, 'feature_fraction': 0.934401506774211,
'bagging_fraction': 0.9909188883235162, 'bagging_freq': 2, 'min_child_samples':
30, 'max_depth': 16}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:25,026] Trial 31 finished with value: 0.02532410147437365
and parameters: {'lambda_l1': 2.0791306227378088e-07, 'lambda_l2':
0.3976867721707133, 'num_leaves': 225, 'feature_fraction': 0.9960318175332967,
'bagging_fraction': 0.7238016789506951, 'bagging_freq': 2, 'min_child_samples':
14, 'max_depth': 13}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:27,243] Trial 32 finished with value: 0.025333524091141583
and parameters: {'lambda_l1': 4.167183982675332e-07, 'lambda_l2':
0.2440827365412648, 'num_leaves': 203, 'feature_fraction': 0.942892282838741,
'bagging_fraction': 0.6297444980378251, 'bagging_freq': 3, 'min_child_samples':
20, 'max_depth': 13}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:29,976] Trial 33 finished with value: 0.025346820963411926
and parameters: {'lambda_l1': 5.3820385522982284e-08, 'lambda_l2':
2.595084829568406, 'num_leaves': 242, 'feature_fraction': 0.9843551972265701,
'bagging_fraction': 0.7690763536570353, 'bagging_freq': 1, 'min_child_samples':
10, 'max_depth': 17}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:32,026] Trial 34 finished with value: 0.02545318502691555
and parameters: {'lambda_l1': 1.0750113161601342e-07, 'lambda_l2':
0.027125809387876867, 'num_leaves': 230, 'feature_fraction': 0.880690644078995,
'bagging_fraction': 0.8817186045605635, 'bagging_freq': 2, 'min_child_samples':
28, 'max_depth': 9}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:34,559] Trial 35 finished with value: 0.02534460382494952
and parameters: {'lambda_l1': 3.867378062079755e-06, 'lambda_l2':
0.09815228483600108, 'num_leaves': 150, 'feature_fraction': 0.8099267257555833,
'bagging_fraction': 0.6905493108348186, 'bagging_freq': 1, 'min_child_samples':
20, 'max_depth': 12}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:37,409] Trial 36 finished with value: 0.025330956782528908
and parameters: {'lambda_l1': 2.0328964968786917e-08, 'lambda_l2':
0.4158455696685313, 'num_leaves': 207, 'feature_fraction': 0.9319402027264925,
'bagging_fraction': 0.6596965173585956, 'bagging_freq': 5, 'min_child_samples':
11, 'max_depth': 15}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:39,026] Trial 37 finished with value: 0.025372546671942525
and parameters: {'lambda_l1': 3.544078368676739e-08, 'lambda_l2':
8.692569615684034, 'num_leaves': 234, 'feature_fraction': 0.8636805069688335,
'bagging_fraction': 0.8150883711323692, 'bagging_freq': 0, 'min_child_samples':
37, 'max_depth': 21}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:42,243] Trial 38 finished with value: 0.02571503039346624
and parameters: {'lambda_l1': 4.775297340749443e-07, 'lambda_l2':
0.0034947307784020474, 'num_leaves': 175, 'feature_fraction':
0.9591463621142414, 'bagging_fraction': 0.7298487699604026, 'bagging_freq': 1,
'min_child_samples': 19, 'max_depth': 14}. Best is trial 21 with value:
0.02530604193384279.
[I 2023-12-13 10:14:45,210] Trial 39 finished with value: 0.025338647674945633
and parameters: {'lambda_l1': 1.0360454335339668e-07, 'lambda_l2':
```

```
2.204208043590473, 'num_leaves': 194, 'feature_fraction': 0.8105533723777831,
'bagging_fraction': 0.9332262956529958, 'bagging_freq': 2, 'min_child_samples':
10, 'max_depth': 30}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:48,943] Trial 40 finished with value: 0.02536484519010719
and parameters: {'lambda_l1': 2.0020771608643243e-06, 'lambda_l2':
0.10013281114432705, 'num_leaves': 256, 'feature_fraction': 0.9994066506813124,
'bagging_fraction': 0.8572564288714862, 'bagging_freq': 5, 'min_child_samples':
5, 'max_depth': 11}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:51,543] Trial 41 finished with value: 0.02532118102788905
and parameters: {'lambda_l1': 1.903617042396143e-07, 'lambda_l2':
0.5422440030625237, 'num_leaves': 223, 'feature_fraction': 0.9932211717700297,
'bagging_fraction': 0.7100489638189486, 'bagging_freq': 2, 'min_child_samples':
15, 'max_depth': 14}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:54,160] Trial 42 finished with value: 0.025322583210416263
and parameters: {'lambda_l1': 3.507525938943139e-08, 'lambda_l2':
0.47056138941543424, 'num_leaves': 228, 'feature_fraction': 0.9558345752004163,
'bagging_fraction': 0.6732294507630332, 'bagging_freq': 3, 'min_child_samples':
14, 'max_depth': 17}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:56,009] Trial 43 finished with value: 0.025329258848715286
and parameters: {'lambda_l1': 2.8964183248924867e-08, 'lambda_l2':
0.5394108999705148, 'num_leaves': 229, 'feature_fraction': 0.9428705464627473,
'bagging_fraction': 0.670706825811867, 'bagging_freq': 3, 'min_child_samples':
28, 'max_depth': 17}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:14:58,009] Trial 44 finished with value: 0.025363425048295803
and parameters: {'lambda_l1': 1.0531640828257046e-05, 'lambda_l2':
3.6963126473623213, 'num_leaves': 242, 'feature_fraction': 0.9589916374013185,
'bagging_fraction': 0.6152781340402177, 'bagging_freq': 3, 'min_child_samples':
14, 'max_depth': 19}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:00,313] Trial 45 finished with value: 0.02532520831644866
and parameters: {'lambda_l1': 1.0324995067227782e-08, 'lambda_l2':
0.315330301709243, 'num_leaves': 187, 'feature_fraction': 0.8549097991558126,
'bagging_fraction': 0.6458979485576598, 'bagging_freq': 4, 'min_child_samples':
21, 'max_depth': 16}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:02,515] Trial 46 finished with value: 0.025408425804381567
and parameters: {'lambda_l1': 2.97197113483365e-07, 'lambda_l2':
0.03637983206929272, 'num_leaves': 189, 'feature_fraction': 0.8624956308542993,
'bagging_fraction': 0.6751648991221478, 'bagging_freq': 4, 'min_child_samples':
33, 'max_depth': 16}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:04,793] Trial 47 finished with value: 0.025328098758577437
and parameters: {'lambda_l1': 8.557609263185527e-08, 'lambda_l2':
0.2135203506517141, 'num_leaves': 211, 'feature_fraction': 0.824882002417801,
'bagging_fraction': 0.7528667664832528, 'bagging_freq': 4, 'min_child_samples':
23, 'max_depth': 13}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:06,826] Trial 48 finished with value: 0.025554832968918178
and parameters: {'lambda_l1': 7.873738474228998e-07, 'lambda_l2':
0.014235015208314508, 'num_leaves': 177, 'feature_fraction': 0.9020402050755106,
'bagging_fraction': 0.7089985631570148, 'bagging_freq': 6, 'min_child_samples':
18, 'max_depth': 9}. Best is trial 21 with value: 0.02530604193384279.
```

```
[I 2023-12-13 10:15:08,159] Trial 49 finished with value: 0.025349672580663826
and parameters: {'lambda_l1': 2.070318010086598e-08, 'lambda_l2':
1.579620353718937, 'num_leaves': 242, 'feature_fraction': 0.8484508753103377,
'bagging_fraction': 0.6454916504363288, 'bagging_freq': 5, 'min_child_samples':
42, 'max_depth': 18}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:10,393] Trial 50 finished with value: 0.025321672056829767
and parameters: {'lambda_l1': 1.1368980457773724e-07, 'lambda_l2':
0.5428674143658029, 'num_leaves': 157, 'feature_fraction': 0.9599035897675225,
'bagging_fraction': 0.7976992661597962, 'bagging_freq': 7, 'min_child_samples':
23, 'max_depth': 14}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:12,516] Trial 51 finished with value: 0.025334096283247284
and parameters: {'lambda_l1': 1.2873246030093563e-07, 'lambda_l2':
0.5599222098672434, 'num_leaves': 143, 'feature_fraction': 0.9630351651756597,
'bagging_fraction': 0.7917497244674868, 'bagging_freq': 7, 'min_child_samples':
25, 'max_depth': 15}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:15,426] Trial 52 finished with value: 0.02534033502098199
and parameters: {'lambda_l1': 3.6849607103874476e-08, 'lambda_l2':
0.07826927409319316, 'num_leaves': 165, 'feature_fraction': 0.916878586208334,
'bagging_fraction': 0.7298979275444871, 'bagging_freq': 4, 'min_child_samples':
12, 'max_depth': 11}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:17,476] Trial 53 finished with value: 0.025324919945201637
and parameters: {'lambda_l1': 6.666337815547867e-07, 'lambda_l2':
0.24549395904259586, 'num_leaves': 187, 'feature_fraction': 0.9631275992898638,
'bagging_fraction': 0.790156705592359, 'bagging_freq': 8, 'min_child_samples':
32, 'max_depth': 16}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:19,192] Trial 54 finished with value: 0.025355823590083548
and parameters: {'lambda_l1': 1.968156031483972e-06, 'lambda_l2':
3.600741100950345, 'num_leaves': 117, 'feature_fraction': 0.964724505204617,
'bagging_fraction': 0.7936723422591964, 'bagging_freq': 8, 'min_child_samples':
29, 'max_depth': 13}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:20,278] Trial 55 finished with value: 0.02534694698852711
and parameters: {'lambda_l1': 3.8377111225234706e-07, 'lambda_l2':
0.18904951933044103, 'num_leaves': 160, 'feature_fraction': 0.9691048942066153,
'bagging_fraction': 0.7654891871882377, 'bagging_freq': 8, 'min_child_samples':
35, 'max_depth': 6}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:22,743] Trial 56 finished with value: 0.02532960234709058
and parameters: {'lambda_l1': 8.531188256993356e-07, 'lambda_l2':
0.7526614331964334, 'num_leaves': 226, 'feature_fraction': 0.8835381046107534,
'bagging_fraction': 0.7051749718794282, 'bagging_freq': 7, 'min_child_samples':
17, 'max_depth': 14}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:24,475] Trial 57 finished with value: 0.025438936531604582
and parameters: {'lambda_l1': 2.401488337081086e-07, 'lambda_l2':
0.040409265626095356, 'num_leaves': 176, 'feature_fraction': 0.932545286477699,
'bagging_fraction': 0.7587957367384308, 'bagging_freq': 7, 'min_child_samples':
54, 'max_depth': 12}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:27,349] Trial 58 finished with value: 0.025331764742036793
and parameters: {'lambda_l1': 7.491210972231241e-06, 'lambda_l2':
1.1264037066318613, 'num_leaves': 191, 'feature_fraction': 0.9768779229557619,
```

'bagging_fraction': 0.8233010761001321, 'bagging_freq': 7, 'min_child_samples': 9, 'max_depth': 17}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:29,126] Trial 59 finished with value: 0.025364522635957107 and parameters: {'lambda_l1': 1.542838500863063e-07, 'lambda_l2': 4.877588286521514, 'num_leaves': 214, 'feature_fraction': 0.9067281575921089, 'bagging_fraction': 0.7439196296824628, 'bagging_freq': 8, 'min_child_samples': 23, 'max_depth': 15}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:30,759] Trial 60 finished with value: 0.02533593828518759 and parameters: {'lambda_l1': 6.369859113896354e-08, 'lambda_l2': 2.079236656483696, 'num_leaves': 244, 'feature_fraction': 0.9425619573733167, 'bagging_fraction': 0.6985206197239799, 'bagging_freq': 0, 'min_child_samples': 26, 'max_depth': 9}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:33,025] Trial 61 finished with value: 0.02532644190312098 and parameters: {'lambda_l1': 1.0624612372376354e-08, 'lambda_l2': 0.38393377183170835, 'num_leaves': 184, 'feature_fraction': 0.9980975123679953, 'bagging_fraction': 0.6677520618526147, 'bagging_freq': 3, 'min_child_samples': 20, 'max_depth': 18}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:35,761] Trial 62 finished with value: 0.02531800140414141 and parameters: {'lambda_l1': 2.7956864358939194e-08, 'lambda_l2': 0.2836592872558351, 'num_leaves': 200, 'feature_fraction': 0.8757525915565615, 'bagging_fraction': 0.7217829817356165, 'bagging_freq': 3, 'min_child_samples': 14, 'max_depth': 16}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:38,843] Trial 63 finished with value: 0.025341272463301798 and parameters: {'lambda_l1': 2.4124032303024824e-08, 'lambda_l2': 0.12877137444280606, 'num_leaves': 200, 'feature_fraction': 0.9720938813526213, 'bagging_fraction': 0.7223197400472539, 'bagging_freq': 2, 'min_child_samples': 14, 'max_depth': 16}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:41,209] Trial 64 finished with value: 0.025329463473603535 and parameters: {'lambda_l1': 1.267753098019513e-07, 'lambda_l2': 0.7476154243852382, 'num_leaves': 210, 'feature_fraction': 0.8896334261190563, 'bagging_fraction': 0.790790950160188, 'bagging_freq': 3, 'min_child_samples': 16, 'max_depth': 12}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:44,960] Trial 65 finished with value: 0.025372393917748842 and parameters: {'lambda_l1': 6.155946908011764e-08, 'lambda_l2': 0.06931548987998583, 'num_leaves': 219, 'feature_fraction': 0.9208928429881679, 'bagging_fraction': 0.7400406535134788, 'bagging_freq': 8, 'min_child_samples': 8, 'max_depth': 14}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:46,108] Trial 66 finished with value: 0.025347768058544023 and parameters: {'lambda_l1': 6.103290232702674e-07, 'lambda_l2': 0.2779033579497487, 'num_leaves': 20, 'feature_fraction': 0.9529225181332778, 'bagging_fraction': 0.6829530919768956, 'bagging_freq': 3, 'min_child_samples': 12, 'max_depth': 19}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:47,693] Trial 67 finished with value: 0.025340558158266926 and parameters: {'lambda_l1': 2.400910251256847e-07, 'lambda_l2': 1.3796844050373616, 'num_leaves': 139, 'feature_fraction': 0.9838492579417322, 'bagging_fraction': 0.777987933114922, 'bagging_freq': 2, 'min_child_samples': 32, 'max_depth': 10}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:49,100] Trial 68 finished with value: 0.025417690099382932

```
and parameters: {'lambda_l1': 4.3779715450034836e-08, 'lambda_l2':
0.02681178446348484, 'num_leaves': 227, 'feature_fraction': 0.9163743144080774,
'bagging_fraction': 0.7091739779099172, 'bagging_freq': 0, 'min_child_samples':
46, 'max_depth': 8}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:51,812] Trial 69 finished with value: 0.025323415780543347
and parameters: {'lambda_l1': 1.919201060718963e-08, 'lambda_l2':
0.5678896095936768, 'num_leaves': 234, 'feature_fraction': 0.8788304237982927,
'bagging_fraction': 0.8334878956046805, 'bagging_freq': 3, 'min_child_samples':
17, 'max_depth': 15}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:53,512] Trial 70 finished with value: 0.025372583980473526
and parameters: {'lambda_l1': 1.9100305067720298e-08, 'lambda_l2':
4.997142902072697, 'num_leaves': 237, 'feature_fraction': 0.7800612671081523,
'bagging_fraction': 0.6134005375827545, 'bagging_freq': 3, 'min_child_samples':
18, 'max_depth': 13}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:56,676] Trial 71 finished with value: 0.02532735757067093
and parameters: {'lambda_l1': 7.069842809463627e-08, 'lambda_l2':
0.5190235961060478, 'num_leaves': 251, 'feature_fraction': 0.9441444932425853,
'bagging_fraction': 0.8273076231109333, 'bagging_freq': 2, 'min_child_samples':
14, 'max_depth': 15}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:15:59,298] Trial 72 finished with value: 0.02532461250649913
and parameters: {'lambda_l1': 3.269935914623592e-08, 'lambda_l2':
0.16572704568230925, 'num_leaves': 215, 'feature_fraction': 0.8736235691393193,
'bagging_fraction': 0.8039882479770437, 'bagging_freq': 3, 'min_child_samples':
23, 'max_depth': 17}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:03,109] Trial 73 finished with value: 0.025332433623973003
and parameters: {'lambda_l1': 1.7968346744935934e-08, 'lambda_l2':
0.15483894369209178, 'num_leaves': 234, 'feature_fraction': 0.8312919180976461,
'bagging_fraction': 0.8387771930988366, 'bagging_freq': 3, 'min_child_samples':
8, 'max_depth': 21}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:05,228] Trial 74 finished with value: 0.02534188139400153
and parameters: {'lambda_l1': 3.9317138948627646e-08, 'lambda_l2':
1.6446840785751544, 'num_leaves': 217, 'feature_fraction': 0.8730552501620695,
'bagging_fraction': 0.8058485381676116, 'bagging_freq': 4, 'min_child_samples':
22, 'max_depth': 18}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:07,975] Trial 75 finished with value: 0.025326018379353138
and parameters: {'lambda_l1': 1.242025665046145e-07, 'lambda_l2':
0.7648410161880234, 'num_leaves': 222, 'feature_fraction': 0.8672494726141169,
'bagging_fraction': 0.7720259320615351, 'bagging_freq': 3, 'min_child_samples':
11, 'max_depth': 17}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:10,443] Trial 76 finished with value: 0.025343555936071613
and parameters: {'lambda_l1': 1.7253621663326098e-08, 'lambda_l2':
0.09051012814507008, 'num_leaves': 84, 'feature_fraction': 0.9027272526442044,
'bagging_fraction': 0.8914020340378658, 'bagging_freq': 2, 'min_child_samples':
16, 'max_depth': 14}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:12,895] Trial 77 finished with value: 0.02531814187324544
and parameters: {'lambda_l1': 3.633581986470781e-08, 'lambda_l2':
0.3875523308803436, 'num_leaves': 206, 'feature_fraction': 0.9998134282771016,
'bagging_fraction': 0.848621981398267, 'bagging_freq': 4, 'min_child_samples':
```

```
25, 'max_depth': 20}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:15,342] Trial 78 finished with value: 0.02532589196595455
and parameters: {'lambda_l1': 7.785691790674241e-08, 'lambda_l2':
0.4012084255288548, 'num_leaves': 201, 'feature_fraction': 0.989427922415336,
'bagging_fraction': 0.875145178733262, 'bagging_freq': 5, 'min_child_samples':
26, 'max_depth': 23}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:16,460] Trial 79 finished with value: 0.025345028141010736
and parameters: {'lambda_l1': 1.9436899767525345e-07, 'lambda_l2':
1.105290518687282, 'num_leaves': 105, 'feature_fraction': 0.9402300852028075,
'bagging_fraction': 0.8403366156989237, 'bagging_freq': 4, 'min_child_samples':
92, 'max_depth': 22}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:19,658] Trial 80 finished with value: 0.025362514815093634
and parameters: {'lambda_l1': 3.537641662439742e-07, 'lambda_l2':
0.04781221888918738, 'num_leaves': 207, 'feature_fraction': 0.9775501600785846,
'bagging_fraction': 0.7306505567041004, 'bagging_freq': 4, 'min_child_samples':
18, 'max_depth': 25}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:22,528] Trial 81 finished with value: 0.025323804360751483
and parameters: {'lambda_l1': 3.646181311231277e-08, 'lambda_l2':
0.1623238497713, 'num_leaves': 226, 'feature_fraction': 0.9967309145484113,
'bagging_fraction': 0.8492885216853667, 'bagging_freq': 3, 'min_child_samples':
22, 'max_depth': 20}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:24,059] Trial 82 finished with value: 0.025330480150483163
and parameters: {'lambda_l1': 3.180350411750548e-08, 'lambda_l2':
0.35168595006965747, 'num_leaves': 248, 'feature_fraction': 0.982258721014483,
'bagging_fraction': 0.8494336131343225, 'bagging_freq': 1, 'min_child_samples':
65, 'max_depth': 20}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:25,892] Trial 83 finished with value: 0.025346955863576284
and parameters: {'lambda_l1': 1.4620244690273026e-08, 'lambda_l2':
2.0909059714970746, 'num_leaves': 228, 'feature_fraction': 0.9978619806835649,
'bagging_fraction': 0.7517485029974953, 'bagging_freq': 3, 'min_child_samples':
21, 'max_depth': 11}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:29,093] Trial 84 finished with value: 0.02531596201710104
and parameters: {'lambda_l1': 8.973775346589939e-08, 'lambda_l2':
0.5271717526844476, 'num_leaves': 238, 'feature_fraction': 0.9260296976313825,
'bagging_fraction': 0.8983957370813418, 'bagging_freq': 2, 'min_child_samples':
13, 'max_depth': 20}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:32,193] Trial 85 finished with value: 0.025323033256888932
and parameters: {'lambda_l1': 8.427615515381058e-08, 'lambda_l2':
0.7339778765588877, 'num_leaves': 195, 'feature_fraction': 0.9049953300818462,
'bagging_fraction': 0.9091786711392047, 'bagging_freq': 4, 'min_child_samples':
12, 'max_depth': 24}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:35,319] Trial 86 finished with value: 0.02532934642085163
and parameters: {'lambda_l1': 1.0009393041248428e-08, 'lambda_l2':
1.171340588424184, 'num_leaves': 193, 'feature_fraction': 0.8958522904063538,
'bagging_fraction': 0.8935195847818315, 'bagging_freq': 5, 'min_child_samples':
8, 'max_depth': 22}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:38,658] Trial 87 finished with value: 0.025319717605636848
and parameters: {'lambda_l1': 5.670637811304032e-08, 'lambda_l2':
```

0.6127554237645482, 'num_leaves': 237, 'feature_fraction': 0.9187283427525719, 'bagging_fraction': 0.9166858761806861, 'bagging_freq': 4, 'min_child_samples': 12, 'max_depth': 26}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:41,299] Trial 88 finished with value: 0.025366722093361105 and parameters: {'lambda_l1': 9.408533022049384e-08, 'lambda_l2': 6.901744838339154, 'num_leaves': 206, 'feature_fraction': 0.9250707986547696, 'bagging_fraction': 0.9151758018042983, 'bagging_freq': 4, 'min_child_samples': 12, 'max_depth': 27}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:44,343] Trial 89 finished with value: 0.025344969496390448 and parameters: {'lambda_l1': 5.774764271338262e-08, 'lambda_l2': 2.3907524045077726, 'num_leaves': 239, 'feature_fraction': 0.95033065961813, 'bagging_fraction': 0.9197855316681319, 'bagging_freq': 5, 'min_child_samples': 10, 'max_depth': 25}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:47,449] Trial 90 finished with value: 0.025325074329485147 and parameters: {'lambda_l1': 1.037640586631444e-07, 'lambda_l2': 0.8347411224111294, 'num_leaves': 197, 'feature_fraction': 0.9200739017412121, 'bagging_fraction': 0.954514378798618, 'bagging_freq': 4, 'min_child_samples': 15, 'max_depth': 24}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:50,993] Trial 91 finished with value: 0.02531208717757587 and parameters: {'lambda_l1': 2.2677241707394913e-08, 'lambda_l2': 0.48589810320019833, 'num_leaves': 233, 'feature_fraction': 0.8952755352474522, 'bagging_fraction': 0.868653851625495, 'bagging_freq': 4, 'min_child_samples': 5, 'max_depth': 27}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:54,695] Trial 92 finished with value: 0.02531323903909325 and parameters: {'lambda_l1': 4.6483410989641444e-08, 'lambda_l2': 0.30893993445244483, 'num_leaves': 211, 'feature_fraction': 0.8400206445890217, 'bagging_fraction': 0.8699968537033014, 'bagging_freq': 4, 'min_child_samples': 6, 'max_depth': 29}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:16:58,643] Trial 93 finished with value: 0.02531087656227158 and parameters: {'lambda_l1': 2.7644526883633937e-08, 'lambda_l2': 0.29304650284706424, 'num_leaves': 250, 'feature_fraction': 0.954840926361633, 'bagging_fraction': 0.8765439326134263, 'bagging_freq': 4, 'min_child_samples': 5, 'max_depth': 29}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:17:02,576] Trial 94 finished with value: 0.02531829388988383 and parameters: {'lambda_l1': 2.3625630862462655e-08, 'lambda_l2': 0.24840698650685364, 'num_leaves': 246, 'feature_fraction': 0.9344309023622013, 'bagging_fraction': 0.862881973432871, 'bagging_freq': 5, 'min_child_samples': 5, 'max_depth': 30}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:17:06,593] Trial 95 finished with value: 0.0253121125288209 and parameters: {'lambda_l1': 2.8051096582997596e-08, 'lambda_l2': 0.26991444050676955, 'num_leaves': 248, 'feature_fraction': 0.8409071363705743, 'bagging_fraction': 0.862261791797156, 'bagging_freq': 5, 'min_child_samples': 5, 'max_depth': 30}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:17:10,905] Trial 96 finished with value: 0.025347681789596545 and parameters: {'lambda_l1': 2.5641399265555486e-08, 'lambda_l2': 0.09803512116809712, 'num_leaves': 248, 'feature_fraction': 0.8293004189694111, 'bagging_fraction': 0.8672884527992188, 'bagging_freq': 5, 'min_child_samples': 5, 'max_depth': 30}. Best is trial 21 with value: 0.02530604193384279.

```
[I 2023-12-13 10:17:14,745] Trial 97 finished with value: 0.025326313587399174
and parameters: {'lambda_l1': 5.211188350924419e-08, 'lambda_l2':
0.24546543900497336, 'num_leaves': 256, 'feature_fraction': 0.845151464272809,
'bagging_fraction': 0.8827233794679381, 'bagging_freq': 6, 'min_child_samples':
7, 'max_depth': 29}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:17:19,643] Trial 98 finished with value: 0.025382457751286135
and parameters: {'lambda_l1': 1.5456062600319328e-08, 'lambda_l2':
0.070957295825512, 'num_leaves': 246, 'feature_fraction': 0.8068130916225714,
'bagging_fraction': 0.8687179150778297, 'bagging_freq': 5, 'min_child_samples':
5, 'max_depth': 27}. Best is trial 21 with value: 0.02530604193384279.
[I 2023-12-13 10:17:23,961] Trial 99 finished with value: 0.025302875924636647
and parameters: {'lambda_l1': 2.5101659330408527e-08, 'lambda_l2':
0.2593027943492803, 'num_leaves': 237, 'feature_fraction': 0.8546684623067573,
'bagging_fraction': 0.9376952289340382, 'bagging_freq': 4, 'min_child_samples':
7, 'max_depth': 28}. Best is trial 99 with value: 0.025302875924636647.
[I 2023-12-13 10:17:23,961] A new study created in memory with name: no-
name-7358a2d0-6891-4474-bb32-fb8d89e178b7
[I 2023-12-13 10:17:24,348] Trial 0 finished with value: 0.02536728711963713 and
parameters: {'lambda': 2.348881295853308e-05, 'alpha': 3.6010467344475403,
'colsample_bytree': 0.7855951534491241, 'subsample': 0.6789267873576292,
'learning_rate': 0.039643541684062936, 'n_estimators': 240, 'max_depth': 3,
'bagging_fraction': 0.8929409166199482, 'feature_fraction': 0.6808920093945671,
'bagging_freq': 6, 'min_child_samples': 6}. Best is trial 0 with value:
0.02536728711963713.
[I 2023-12-13 10:17:28,945] Trial 1 finished with value: 0.025449030708061985
and parameters: {'lambda': 5.360294728728285, 'alpha': 0.31044435499483225,
'colsample_bytree': 0.36987128854262097, 'subsample': 0.3454599737656805,
'learning_rate': 0.044846856872152424, 'n_estimators': 374, 'max_depth': 17,
'bagging_fraction': 0.5455560149136927, 'feature_fraction': 0.43298331215843355,
'bagging_freq': 5, 'min_child_samples': 18}. Best is trial 0 with value:
0.02536728711963713.
[I 2023-12-13 10:17:42,675] Trial 2 finished with value: 0.02533587630139431 and
parameters: {'lambda': 4.258943089524393e-06, 'alpha': 1.9826980964985924e-05,
'colsample_bytree': 0.5648559873736287, 'subsample': 0.8281407691144109,
'learning_rate': 0.047938018610088354, 'n_estimators': 563, 'max_depth': 19,
'bagging_fraction': 0.2371603301759982, 'feature_fraction': 0.6860358815211507,
'bagging_freq': 1, 'min_child_samples': 11}. Best is trial 2 with value:
0.02533587630139431.
[I 2023-12-13 10:17:45,398] Trial 3 finished with value: 0.025352250271565284
and parameters: {'lambda': 3.4671276804481113, 'alpha': 4.905556676028774,
'colsample_bytree': 0.846717878493169, 'subsample': 0.4436910153386966,
'learning_rate': 0.028557701661212936, 'n_estimators': 716, 'max_depth': 14,
'bagging_fraction': 0.2976305878758231, 'feature_fraction': 0.5961415280890161,
'bagging_freq': 0, 'min_child_samples': 92}. Best is trial 2 with value:
0.02533587630139431.
[I 2023-12-13 10:18:26,752] Trial 4 finished with value: 0.02643453419921624 and
parameters: {'lambda': 2.133142332373004e-06, 'alpha': 0.009176996354542699,
'colsample_bytree': 0.4493688608715288, 'subsample': 0.6160544169422486,
```

'learning_rate': 0.11387495307522313, 'n_estimators': 266, 'max_depth': 30, 'bagging_fraction': 0.8201062586888916, 'feature_fraction': 0.9515991532513512, 'bagging_freq': 8, 'min_child_samples': 62}. Best is trial 2 with value: 0.02533587630139431.
[I 2023-12-13 10:18:27,496] Trial 5 finished with value: 0.025491634117154466 and parameters: {'lambda': 1.9809253750493907, 'alpha': 6.257956190096665e-08, 'colsample_bytree': 0.3567862899353162, 'subsample': 0.23618183112843047, 'learning_rate': 0.07181276284502022, 'n_estimators': 450, 'max_depth': 9, 'bagging_fraction': 0.8629900073215435, 'feature_fraction': 0.4854026613548714, 'bagging_freq': 2, 'min_child_samples': 57}. Best is trial 2 with value: 0.02533587630139431.
[I 2023-12-13 10:18:27,803] Trial 6 finished with value: 0.025339004723380448 and parameters: {'lambda': 1.8548894229694903e-07, 'alpha': 0.16587190283399655, 'colsample_bytree': 0.2596405149438167, 'subsample': 0.9895095492804138, 'learning_rate': 0.1567265061663649, 'n_estimators': 279, 'max_depth': 2, 'bagging_fraction': 0.8523691427638673, 'feature_fraction': 0.7654858750780937, 'bagging_freq': 6, 'min_child_samples': 79}. Best is trial 2 with value: 0.02533587630139431.
[I 2023-12-13 10:18:28,099] Trial 7 finished with value: 0.02534053570446774 and parameters: {'lambda': 4.638759594322625e-08, 'alpha': 1.683416412018213e-05, 'colsample_bytree': 0.2926952476201038, 'subsample': 0.890482740700475, 'learning_rate': 0.128426644097236, 'n_estimators': 398, 'max_depth': 3, 'bagging_fraction': 0.4487858573725298, 'feature_fraction': 0.46014665762139767, 'bagging_freq': 6, 'min_child_samples': 66}. Best is trial 2 with value: 0.02533587630139431.
[I 2023-12-13 10:18:58,743] Trial 8 finished with value: 0.026668828386364175 and parameters: {'lambda': 0.9658611176861268, 'alpha': 0.0001778010520878397, 'colsample_bytree': 0.29567539675064136, 'subsample': 0.7705958297783961, 'learning_rate': 0.15454915923721052, 'n_estimators': 605, 'max_depth': 24, 'bagging_fraction': 0.5950364770915126, 'feature_fraction': 0.6181862635055952, 'bagging_freq': 3, 'min_child_samples': 7}. Best is trial 2 with value: 0.02533587630139431.
[I 2023-12-13 10:18:59,591] Trial 9 finished with value: 0.025526670767535856 and parameters: {'lambda': 9.354548757337708e-08, 'alpha': 1.9180621318615033e-08, 'colsample_bytree': 0.7091283290110244, 'subsample': 0.45148478486106136, 'learning_rate': 0.10662843132129353, 'n_estimators': 917, 'max_depth': 9, 'bagging_fraction': 0.5283063384285038, 'feature_fraction': 0.804440910834439, 'bagging_freq': 2, 'min_child_samples': 12}. Best is trial 2 with value: 0.02533587630139431.
[I 2023-12-13 10:19:14,961] Trial 10 finished with value: 0.025324266781687428 and parameters: {'lambda': 0.0004197301954670796, 'alpha': 3.189002659385657e-06, 'colsample_bytree': 0.9756296886302929, 'subsample': 0.7887340933444849, 'learning_rate': 0.013263423550739381, 'n_estimators': 809, 'max_depth': 22, 'bagging_fraction': 0.20870376720780998, 'feature_fraction': 0.277575023798624, 'bagging_freq': 0, 'min_child_samples': 31}. Best is trial 10 with value: 0.025324266781687428.
[I 2023-12-13 10:19:30,462] Trial 11 finished with value: 0.025334531961421297 and parameters: {'lambda': 0.0010634634713376783, 'alpha':

```
3.738425892816665e-06, 'colsample_bytree': 0.9482709457781746, 'subsample':
0.7923980072673472, 'learning_rate': 0.012107894436734875, 'n_estimators': 856,
'max_depth': 22, 'bagging_fraction': 0.2062123411650716, 'feature_fraction':
0.20531864177197623, 'bagging_freq': 0, 'min_child_samples': 34}. Best is trial
10 with value: 0.025324266781687428.
[I 2023-12-13 10:19:51,089] Trial 12 finished with value: 0.025342805698212838
and parameters: {'lambda': 0.0026822885374401797, 'alpha':
4.818242369697557e-07, 'colsample_bytree': 0.9913992236985012, 'subsample':
0.7066548131136762, 'learning_rate': 0.010602666715584288, 'n_estimators': 997,
'max_depth': 25, 'bagging_fraction': 0.20236154368969336, 'feature_fraction':
0.20147439461483163, 'bagging_freq': 0, 'min_child_samples': 35}. Best is trial
10 with value: 0.025324266781687428.
[I 2023-12-13 10:20:10,018] Trial 13 finished with value: 0.025317980690404557
and parameters: {'lambda': 0.0013484871468783942, 'alpha':
1.9506464040348895e-06, 'colsample_bytree': 0.9984167770407494, 'subsample':
0.8773892361272996, 'learning_rate': 0.011291595807886915, 'n_estimators': 828,
'max_depth': 22, 'bagging_fraction': 0.3538879028707166, 'feature_fraction':
0.21604267050223241, 'bagging_freq': 0, 'min_child_samples': 36}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:20:59,206] Trial 14 finished with value: 0.02580076854855439
and parameters: {'lambda': 0.02209753057171333, 'alpha': 6.134922565400404e-07,
'colsample_bytree': 0.9978203971727524, 'subsample': 0.9754158816389745,
'learning_rate': 0.08105576409090104, 'n_estimators': 732, 'max_depth': 29,
'bagging_fraction': 0.3490486487775064, 'feature_fraction': 0.3132673933218812,
'bagging_freq': 3, 'min_child_samples': 35}. Best is trial 13 with value:
0.025317980690404557.
[I 2023-12-13 10:21:15,214] Trial 15 finished with value: 0.02552926740251587
and parameters: {'lambda': 0.00010314440090791428, 'alpha':
0.0003040911323711148, 'colsample_bytree': 0.8788824525560035, 'subsample':
0.8724297533826497, 'learning_rate': 0.06899309222525574, 'n_estimators': 761,
'max_depth': 21, 'bagging_fraction': 0.3799389355509811, 'feature_fraction':
0.33557862011876527, 'bagging_freq': 1, 'min_child_samples': 46}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:21:18,918] Trial 16 finished with value: 0.026319866122877084
and parameters: {'lambda': 0.015320050214655337, 'alpha':
2.4922306151320825e-07, 'colsample_bytree': 0.7276315712072438, 'subsample':
0.722967692675315, 'learning_rate': 0.1990136150267524, 'n_estimators': 842,
'max_depth': 14, 'bagging_fraction': 0.4112562931698962, 'feature_fraction':
0.3376101836055925, 'bagging_freq': 1, 'min_child_samples': 24}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:21:47,773] Trial 17 finished with value: 0.02532257905306865
and parameters: {'lambda': 0.00020826825850234454, 'alpha':
1.2155765880428766e-08, 'colsample_bytree': 0.906966034733289, 'subsample':
0.8980328843877672, 'learning_rate': 0.010388075720414449, 'n_estimators': 111,
'max_depth': 26, 'bagging_fraction': 0.28599760844664923, 'feature_fraction':
0.26819890475802455, 'bagging_freq': 4, 'min_child_samples': 44}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:22:19,155] Trial 18 finished with value: 0.025443135909912325
```

```
and parameters: {'lambda': 5.799449092292789e-05, 'alpha':
1.7150556329397997e-08, 'colsample_bytree': 0.8879510049644385, 'subsample':
0.941113187228799, 'learning_rate': 0.05711304652835883, 'n_estimators': 103,
'max_depth': 26, 'bagging_fraction': 0.3021992992655992, 'feature_fraction':
0.39936962241818347, 'bagging_freq': 8, 'min_child_samples': 47}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:22:55,373] Trial 19 finished with value: 0.02533874071798422
and parameters: {'lambda': 1.0135663854176074e-08, 'alpha':
1.098349512211046e-07, 'colsample_bytree': 0.8096046519846736, 'subsample':
0.899321312673695, 'learning_rate': 0.027472124267913893, 'n_estimators': 122,
'max_depth': 27, 'bagging_fraction': 0.4595581452181082, 'feature_fraction':
0.2636043349504611, 'bagging_freq': 4, 'min_child_samples': 46}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:23:09,751] Trial 20 finished with value: 0.025544937786006135
and parameters: {'lambda': 0.004571763094333716, 'alpha':
3.1436322789336505e-08, 'colsample_bytree': 0.651547223596006, 'subsample':
0.9907689499798574, 'learning_rate': 0.08195207526064507, 'n_estimators': 643,
'max_depth': 19, 'bagging_fraction': 0.6620731732309565, 'feature_fraction':
0.39258255945519815, 'bagging_freq': 5, 'min_child_samples': 79}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:23:32,219] Trial 21 finished with value: 0.02532741096942643
and parameters: {'lambda': 0.0005018271886285479, 'alpha':
2.621374255364848e-06, 'colsample_bytree': 0.9272993830839806, 'subsample':
0.8339773340492492, 'learning_rate': 0.014363114737925148, 'n_estimators': 981,
'max_depth': 23, 'bagging_fraction': 0.2736288074876529, 'feature_fraction':
0.2720271539895085, 'bagging_freq': 2, 'min_child_samples': 26}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:24:09,364] Trial 22 finished with value: 0.025324379794827247
and parameters: {'lambda': 0.0002622454964083763, 'alpha':
1.368166134357462e-08, 'colsample_bytree': 0.9286718815684943, 'subsample':
0.9018735990075679, 'learning_rate': 0.029587943386857895, 'n_estimators': 823,
'max_depth': 27, 'bagging_fraction': 0.33449193411689365, 'feature_fraction':
0.2003108304413971, 'bagging_freq': 0, 'min_child_samples': 39}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:24:21,938] Trial 23 finished with value: 0.02534438917106144
and parameters: {'lambda': 0.0005620540652969066, 'alpha':
1.3650352037705418e-06, 'colsample_bytree': 0.983837778960803, 'subsample':
0.7776953982235933, 'learning_rate': 0.010440081281319178, 'n_estimators': 484,
'max_depth': 20, 'bagging_fraction': 0.2922265123542156, 'feature_fraction':
0.28036331188736974, 'bagging_freq': 3, 'min_child_samples': 27}. Best is trial
13 with value: 0.025317980690404557.
[I 2023-12-13 10:24:29,285] Trial 24 finished with value: 0.02535116796323398
and parameters: {'lambda': 2.120833581868303e-05, 'alpha':
1.5053211300309977e-07, 'colsample_bytree': 0.8714621382476472, 'subsample':
0.8459264116842012, 'learning_rate': 0.03658775847837228, 'n_estimators': 695,
'max_depth': 17, 'bagging_fraction': 0.25306227525229374, 'feature_fraction':
0.38198598459170685, 'bagging_freq': 1, 'min_child_samples': 52}. Best is trial
13 with value: 0.025317980690404557.
```

```
[I 2023-12-13 10:24:51,225] Trial 25 finished with value: 0.02528372802431292
and parameters: {'lambda': 0.058810989995884466, 'alpha':
1.1996865537747812e-05, 'colsample_bytree': 0.9228600517854186, 'subsample':
0.92774701821838, 'learning_rate': 0.026063269064339138, 'n_estimators': 925,
'max_depth': 23, 'bagging_fraction': 0.36053695959137383, 'feature_fraction':
0.24232935008224465, 'bagging_freq': 7, 'min_child_samples': 19}. Best is trial
25 with value: 0.02528372802431292.
[I 2023-12-13 10:25:32,656] Trial 26 finished with value: 0.02539940261684259
and parameters: {'lambda': 0.08978561797373089, 'alpha': 2.8982827488604278e-05,
'colsample_bytree': 0.8039872735000635, 'subsample': 0.9329853605841968,
'learning_rate': 0.050152918082088195, 'n_estimators': 922, 'max_depth': 28,
'bagging_fraction': 0.3709929069677738, 'feature_fraction': 0.3464418421089758,
'bagging_freq': 7, 'min_child_samples': 16}. Best is trial 25 with value:
0.02528372802431292.
[I 2023-12-13 10:26:04,093] Trial 27 finished with value: 0.025282512933134216
and parameters: {'lambda': 0.14790769096401055, 'alpha': 2.4893316633915135e-07,
'colsample_bytree': 0.901403347552969, 'subsample': 0.9253059914860593,
'learning_rate': 0.026511745145132506, 'n_estimators': 931, 'max_depth': 25,
'bagging_fraction': 0.44233841213545827, 'feature_fraction': 0.5061274743817828,
'bagging_freq': 7, 'min_child_samples': 22}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:26:34,251] Trial 28 finished with value: 0.02553139328569936
and parameters: {'lambda': 0.07238489034061557, 'alpha': 7.583369278727293e-07,
'colsample_bytree': 0.8413121645010125, 'subsample': 0.9526835088512151,
'learning_rate': 0.056948448620533304, 'n_estimators': 937, 'max_depth': 24,
'bagging_fraction': 0.43454777731213423, 'feature_fraction': 0.5095075827176941,
'bagging_freq': 7, 'min_child_samples': 22}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:26:38,465] Trial 29 finished with value: 0.02534945466510938
and parameters: {'lambda': 0.47848908473429197, 'alpha': 1.4038659728922328e-07,
'colsample_bytree': 0.7629633298773877, 'subsample': 0.6235462112018371,
'learning_rate': 0.03572890209768088, 'n_estimators': 881, 'max_depth': 15,
'bagging_fraction': 0.4724955745579422, 'feature_fraction': 0.5109027039022014,
'bagging_freq': 7, 'min_child_samples': 18}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:26:40,763] Trial 30 finished with value: 0.025333660950016004
and parameters: {'lambda': 0.1822477708912448, 'alpha': 7.387997602911804e-05,
'colsample_bytree': 0.9434391457901615, 'subsample': 0.7327878194698516,
'learning_rate': 0.024771778238551343, 'n_estimators': 774, 'max_depth': 12,
'bagging_fraction': 0.3831433548903835, 'feature_fraction': 0.38867650413097454,
'bagging_freq': 5, 'min_child_samples': 6}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:27:08,082] Trial 31 finished with value: 0.025287884394781017
and parameters: {'lambda': 0.005375611843988055, 'alpha': 6.864223173736183e-06,
'colsample_bytree': 0.9029726366774973, 'subsample': 0.9199379167045219,
'learning_rate': 0.023126940558873946, 'n_estimators': 205, 'max_depth': 25,
'bagging_fraction': 0.3220191404676403, 'feature_fraction': 0.24577684284630236,
'bagging_freq': 4, 'min_child_samples': 42}. Best is trial 27 with value:
```

```
0.025282512933134216.
[I 2023-12-13 10:27:39,550] Trial 32 finished with value: 0.025431346568253684
and parameters: {'lambda': 0.015025322376796317, 'alpha':
7.5035483793019745e-06, 'colsample_bytree': 0.8353827567771035, 'subsample':
0.9985864630609104, 'learning_rate': 0.04433107677834941, 'n_estimators': 191,
'max_depth': 24, 'bagging_fraction': 0.35929261760926257, 'feature_fraction':
0.4223464251147666, 'bagging_freq': 8, 'min_child_samples': 29}. Best is trial
27 with value: 0.025282512933134216.
[I 2023-12-13 10:27:48,184] Trial 33 finished with value: 0.02529896699176874
and parameters: {'lambda': 7.587853897666754, 'alpha': 6.897904921269376e-06,
'colsample_bytree': 0.9079481058657444, 'subsample': 0.8582518235544911,
'learning_rate': 0.025639044071582476, 'n_estimators': 347, 'max_depth': 18,
'bagging_fraction': 0.32638254406050576, 'feature_fraction':
0.24089963286515217, 'bagging_freq': 7, 'min_child_samples': 40}. Best is trial
27 with value: 0.025282512933134216.
[I 2023-12-13 10:27:56,284] Trial 34 finished with value: 0.02535826414098982
and parameters: {'lambda': 5.264696606328038, 'alpha': 1.1362660840789599e-05,
'colsample_bytree': 0.8913601953606834, 'subsample': 0.8303034339798494,
'learning_rate': 0.039578899716257554, 'n_estimators': 339, 'max_depth': 18,
'bagging_fraction': 0.40804852483635956, 'feature_fraction': 0.3338583098277596,
'bagging_freq': 7, 'min_child_samples': 20}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:28:10,338] Trial 35 finished with value: 0.02532723030722082
and parameters: {'lambda': 0.40998405457144194, 'alpha': 5.194852245588099e-05,
'colsample_bytree': 0.7959732006999652, 'subsample': 0.9501828002607869,
'learning_rate': 0.02669405788864749, 'n_estimators': 188, 'max_depth': 20,
'bagging_fraction': 0.31529221715040284, 'feature_fraction': 0.4421756651386456,
'bagging_freq': 5, 'min_child_samples': 13}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:29:01,923] Trial 36 finished with value: 0.025435242909906793
and parameters: {'lambda': 6.070543611683524, 'alpha': 7.0963414124749754e-06,
'colsample_bytree': 0.841173188976618, 'subsample': 0.925089632534882,
'learning_rate': 0.047640048323690634, 'n_estimators': 329, 'max_depth': 30,
'bagging_fraction': 0.4935299149388039, 'feature_fraction': 0.23311014810517147,
'bagging_freq': 6, 'min_child_samples': 41}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:29:09,229] Trial 37 finished with value: 0.02531323681495283
and parameters: {'lambda': 2.580734328185925, 'alpha': 0.0031038336361912033,
'colsample_bytree': 0.9253764750364106, 'subsample': 0.8463360531777626,
'learning_rate': 0.024885383516498924, 'n_estimators': 524, 'max_depth': 17,
'bagging_fraction': 0.26573350451151395, 'feature_fraction': 0.3127790350395384,
'bagging_freq': 8, 'min_child_samples': 54}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:29:29,039] Trial 38 finished with value: 0.025370003421854234
and parameters: {'lambda': 8.583516401044939, 'alpha': 1.954426827890366e-05,
'colsample_bytree': 0.7666638305032004, 'subsample': 0.6706043725700467,
'learning_rate': 0.037005314181319915, 'n_estimators': 221, 'max_depth': 25,
'bagging_fraction': 0.41719613646128784, 'feature_fraction':
```

0.24077286904507264, 'bagging_freq': 6, 'min_child_samples': 66}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:29:31,483] Trial 39 finished with value: 0.025345419519489654 and parameters: {'lambda': 0.49445759005253825, 'alpha': 0.00010973183167484691, 'colsample_bytree': 0.8672020361371833, 'subsample': 0.9310851510330245, 'learning_rate': 0.05777965463669941, 'n_estimators': 293, 'max_depth': 11, 'bagging_fraction': 0.24248585701312617, 'feature_fraction': 0.5447835593060134, 'bagging_freq': 6, 'min_child_samples': 94}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:29:48,526] Trial 40 finished with value: 0.025325867414575452 and parameters: {'lambda': 1.292996010123885, 'alpha': 0.0005061194403953367, 'colsample_bytree': 0.5536091578184776, 'subsample': 0.8626735779945631, 'learning_rate': 0.020529014734032916, 'n_estimators': 412, 'max_depth': 20, 'bagging_fraction': 0.3351431872533785, 'feature_fraction': 0.46986138597498645, 'bagging_freq': 7, 'min_child_samples': 60}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:29:56,818] Trial 41 finished with value: 0.025324838177586788 and parameters: {'lambda': 2.512908824787915, 'alpha': 0.001550386644021076, 'colsample_bytree': 0.9453815248474408, 'subsample': 0.8405026907867502, 'learning_rate': 0.02405132863735949, 'n_estimators': 544, 'max_depth': 17, 'bagging_fraction': 0.24513870826620765, 'feature_fraction': 0.2940675993728173, 'bagging_freq': 8, 'min_child_samples': 54}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:30:02,920] Trial 42 finished with value: 0.025322805878299326 and parameters: {'lambda': 2.103135583574258, 'alpha': 4.403746993445655e-05, 'colsample_bytree': 0.9224020146203477, 'subsample': 0.8035075204686015, 'learning_rate': 0.034237252557707605, 'n_estimators': 508, 'max_depth': 16, 'bagging_fraction': 0.31222811307455867, 'feature_fraction': 0.30555535280726925, 'bagging_freq': 8, 'min_child_samples': 51}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:30:12,740] Trial 43 finished with value: 0.02530761782114578 and parameters: {'lambda': 1.0265828606686984, 'alpha': 6.196059800731385e-06, 'colsample_bytree': 0.9102605509456637, 'subsample': 0.8697486525220428, 'learning_rate': 0.0224751967465829, 'n_estimators': 430, 'max_depth': 18, 'bagging_fraction': 0.26900870709507846, 'feature_fraction': 0.24412425850994926, 'bagging_freq': 4, 'min_child_samples': 40}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:30:38,060] Trial 44 finished with value: 0.025303773556226573 and parameters: {'lambda': 0.1430188941835898, 'alpha': 5.16411308346194e-06, 'colsample_bytree': 0.8497388177542335, 'subsample': 0.9693426832040742, 'learning_rate': 0.020026886645017908, 'n_estimators': 444, 'max_depth': 23, 'bagging_fraction': 0.3936812277025951, 'feature_fraction': 0.2408143524614192, 'bagging_freq': 4, 'min_child_samples': 40}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:31:02,487] Trial 45 finished with value: 0.025341232919165547 and parameters: {'lambda': 0.0524969672444234, 'alpha': 2.0058708786222933e-06, 'colsample_bytree': 0.8496953554731784, 'subsample': 0.9717281725845497, 'learning_rate': 0.04365865877850472, 'n_estimators': 354, 'max_depth': 22,

```
'bagging_fraction': 0.39570093711460386, 'feature_fraction':
0.24661114628833844, 'bagging_freq': 5, 'min_child_samples': 31}. Best is trial
27 with value: 0.025282512933134216.
[I 2023-12-13 10:31:34,170] Trial 46 finished with value: 0.025314734267894927
and parameters: {'lambda': 0.23269908992152427, 'alpha': 1.611515204727705e-05,
'colsample_bytree': 0.965676569017409, 'subsample': 0.999656095164172,
'learning_rate': 0.01863530252444772, 'n_estimators': 458, 'max_depth': 24,
'bagging_fraction': 0.4943335517237851, 'feature_fraction': 0.3631953272157497,
'bagging_freq': 3, 'min_child_samples': 14}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:31:34,976] Trial 47 finished with value: 0.025344923052158633
and parameters: {'lambda': 0.005896702702321232, 'alpha':
1.0406627241470713e-06, 'colsample_bytree': 0.9618952058852995, 'subsample':
0.9288397315916691, 'learning_rate': 0.030111494338558937, 'n_estimators': 388,
'max_depth': 6, 'bagging_fraction': 0.4282807849114691, 'feature_fraction':
0.23115331082969517, 'bagging_freq': 4, 'min_child_samples': 75}. Best is trial
27 with value: 0.025282512933134216.
[I 2023-12-13 10:32:11,288] Trial 48 finished with value: 0.025316490999835222
and parameters: {'lambda': 0.031212722413492627, 'alpha':
3.7769914516813474e-07, 'colsample_bytree': 0.8884889131865825, 'subsample':
0.9060885931204238, 'learning_rate': 0.01920820130034489, 'n_estimators': 573,
'max_depth': 28, 'bagging_fraction': 0.322914933687394, 'feature_fraction':
0.3127105556848301, 'bagging_freq': 5, 'min_child_samples': 31}. Best is trial
27 with value: 0.025282512933134216.
[I 2023-12-13 10:32:33,692] Trial 49 finished with value: 0.025289914308978065
and parameters: {'lambda': 0.1371564983647184, 'alpha': 3.4714458974934516e-06,
'colsample_bytree': 0.8359868091048415, 'subsample': 0.8815383936274016,
'learning_rate': 0.032929009749312635, 'n_estimators': 293, 'max_depth': 23,
'bagging_fraction': 0.35865626556318825, 'feature_fraction': 0.3596120542560316,
'bagging_freq': 4, 'min_child_samples': 9}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:33:03,508] Trial 50 finished with value: 0.025288314594923346
and parameters: {'lambda': 0.03716394702972381, 'alpha': 1.0145598932622684e-06,
'colsample_bytree': 0.96399209493982, 'subsample': 0.8093528692659182,
'learning_rate': 0.03862283757784743, 'n_estimators': 244, 'max_depth': 26,
'bagging_fraction': 0.35859234394587725, 'feature_fraction': 0.4219241308038516,
'bagging_freq': 6, 'min_child_samples': 10}. Best is trial 27 with value:
0.025282512933134216.
[I 2023-12-13 10:33:32,280] Trial 51 finished with value: 0.025285151914118183
and parameters: {'lambda': 0.050030472287633715, 'alpha':
3.9077224566439686e-07, 'colsample_bytree': 0.9990962226936917, 'subsample':
0.8048273270021098, 'learning_rate': 0.03292791014685181, 'n_estimators': 243,
'max_depth': 26, 'bagging_fraction': 0.3562316108961916, 'feature_fraction':
0.42308401776829585, 'bagging_freq': 7, 'min_child_samples': 9}. Best is trial
27 with value: 0.025282512933134216.
[I 2023-12-13 10:33:59,865] Trial 52 finished with value: 0.025301870742286115
and parameters: {'lambda': 0.03503303441717783, 'alpha': 3.4017918409553e-07,
'colsample_bytree': 0.9968107027563734, 'subsample': 0.8076652135816409,
```

'learning_rate': 0.03335533206027537, 'n_estimators': 246, 'max_depth': 26, 'bagging_fraction': 0.36398710928457817, 'feature_fraction': 0.41790810612769946, 'bagging_freq': 6, 'min_child_samples': 9}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:34:31,623] Trial 53 finished with value: 0.02533660928177224 and parameters: {'lambda': 0.01179422033637659, 'alpha': 1.1337565952805162e-06, 'colsample_bytree': 0.9619757823671682, 'subsample': 0.7663157542870254, 'learning_rate': 0.04217441944923794, 'n_estimators': 156, 'max_depth': 28, 'bagging_fraction': 0.44320756679912193, 'feature_fraction': 0.4492492972778519, 'bagging_freq': 7, 'min_child_samples': 9}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:34:58,917] Trial 54 finished with value: 0.025396756341851217 and parameters: {'lambda': 0.09237888807800435, 'alpha': 5.396006990565863e-08, 'colsample_bytree': 0.9561834646809306, 'subsample': 0.8881725371385112, 'learning_rate': 0.051002919704229224, 'n_estimators': 309, 'max_depth': 25, 'bagging_fraction': 0.35101431913383385, 'feature_fraction': 0.3682346238997096, 'bagging_freq': 6, 'min_child_samples': 5}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:35:43,830] Trial 55 finished with value: 0.0253035237648634 and parameters: {'lambda': 0.043722782227706716, 'alpha': 3.3105036451144713e-06, 'colsample_bytree': 0.886525249892589, 'subsample': 0.8165176884348425, 'learning_rate': 0.032299842957129905, 'n_estimators': 261, 'max_depth': 30, 'bagging_fraction': 0.296991358282764, 'feature_fraction': 0.4290733347567242, 'bagging_freq': 5, 'min_child_samples': 11}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:36:20,334] Trial 56 finished with value: 0.02528500783293694 and parameters: {'lambda': 0.005747101312468892, 'alpha': 8.141560300389794e-07, 'colsample_bytree': 0.9737165090660413, 'subsample': 0.8807436940088339, 'learning_rate': 0.016618968065034195, 'n_estimators': 211, 'max_depth': 27, 'bagging_fraction': 0.38085558911080825, 'feature_fraction': 0.4769190982319051, 'bagging_freq': 4, 'min_child_samples': 17}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:36:52,342] Trial 57 finished with value: 0.025319315732729487 and parameters: {'lambda': 0.006484107144774403, 'alpha': 5.088453360539683e-07, 'colsample_bytree': 0.9950585271782662, 'subsample': 0.7623603942107702, 'learning_rate': 0.01608741728618196, 'n_estimators': 212, 'max_depth': 29, 'bagging_fraction': 0.38810751326610854, 'feature_fraction': 0.48309647031267866, 'bagging_freq': 7, 'min_child_samples': 18}. Best is trial 27 with value: 0.025282512933134216.
[I 2023-12-13 10:37:29,485] Trial 58 finished with value: 0.025279582791079294 and parameters: {'lambda': 0.002523588391891515, 'alpha': 2.0278583823243968e-07, 'colsample_bytree': 0.9683349269055324, 'subsample': 0.9111148733420981, 'learning_rate': 0.015664809123351116, 'n_estimators': 161, 'max_depth': 27, 'bagging_fraction': 0.23157067832807865, 'feature_fraction': 0.5786596532202108, 'bagging_freq': 3, 'min_child_samples': 23}. Best is trial 58 with value: 0.025279582791079294.
[I 2023-12-13 10:38:04,633] Trial 59 finished with value: 0.0253070132425188 and parameters: {'lambda': 0.0019107207585035936, 'alpha': 6.267743818256142e-08,

```
'colsample_bytree': 0.9419808879994382, 'subsample': 0.9077285825160895,
'learning_rate': 0.010299746992342328, 'n_estimators': 151, 'max_depth': 27,
'bagging_fraction': 0.22512925990724408, 'feature_fraction': 0.5814847795550019,
'bagging_freq': 2, 'min_child_samples': 23}. Best is trial 58 with value:
0.025279582791079294.
[I 2023-12-13 10:38:52,863] Trial 60 finished with value: 0.02528462266795434
and parameters: {'lambda': 0.003138580147412608, 'alpha': 1.86695368039163e-07,
'colsample_bytree': 0.9737654053577157, 'subsample': 0.9688230202499536,
'learning_rate': 0.01735639197034104, 'n_estimators': 142, 'max_depth': 29,
'bagging_fraction': 0.21712131021426453, 'feature_fraction': 0.6325220550193417,
'bagging_freq': 3, 'min_child_samples': 15}. Best is trial 58 with value:
0.025279582791079294.
[I 2023-12-13 10:39:38,928] Trial 61 finished with value: 0.02528742250218979
and parameters: {'lambda': 0.0037226844116723533, 'alpha':
2.4955257667563724e-07, 'colsample_bytree': 0.9129202552882323, 'subsample':
0.9580388131060794, 'learning_rate': 0.02899862090694192, 'n_estimators': 148,
'max_depth': 29, 'bagging_fraction': 0.23000449355222463, 'feature_fraction':
0.6418874937646275, 'bagging_freq': 3, 'min_child_samples': 16}. Best is trial
58 with value: 0.025279582791079294.
[I 2023-12-13 10:40:24,066] Trial 62 finished with value: 0.025279323598135478
and parameters: {'lambda': 0.0010253094992266239, 'alpha':
1.493324353314564e-07, 'colsample_bytree': 0.9778727383726121, 'subsample':
0.9585818132592492, 'learning_rate': 0.017037717674944206, 'n_estimators': 150,
'max_depth': 29, 'bagging_fraction': 0.20357196280730727, 'feature_fraction':
0.6429256667992529, 'bagging_freq': 3, 'min_child_samples': 15}. Best is trial
62 with value: 0.025279323598135478.
[I 2023-12-13 10:41:10,917] Trial 63 finished with value: 0.025283259657986647
and parameters: {'lambda': 0.001102031892047964, 'alpha':
1.1722490281808559e-07, 'colsample_bytree': 0.9732176069331557, 'subsample':
0.9644847112263495, 'learning_rate': 0.01770557138239636, 'n_estimators': 169,
'max_depth': 29, 'bagging_fraction': 0.20610686098273098, 'feature_fraction':
0.5837076469660731, 'bagging_freq': 3, 'min_child_samples': 21}. Best is trial
62 with value: 0.025279323598135478.
[I 2023-12-13 10:41:57,640] Trial 64 finished with value: 0.025286118527907082
and parameters: {'lambda': 0.0010522006592077015, 'alpha':
1.0307512265253673e-07, 'colsample_bytree': 0.9764059851924647, 'subsample':
0.9690772399115555, 'learning_rate': 0.016537790197938492, 'n_estimators': 176,
'max_depth': 29, 'bagging_fraction': 0.20830537327303003, 'feature_fraction':
0.6686680793441417, 'bagging_freq': 2, 'min_child_samples': 21}. Best is trial
62 with value: 0.025279323598135478.
[I 2023-12-13 10:42:31,134] Trial 65 finished with value: 0.025319389033990156
and parameters: {'lambda': 0.0024312132163255033, 'alpha':
3.452982173209635e-08, 'colsample_bytree': 0.937565535150826, 'subsample':
0.9516964016489521, 'learning_rate': 0.014214572628491738, 'n_estimators': 119,
'max_depth': 27, 'bagging_fraction': 0.20203007289339245, 'feature_fraction':
0.5990342542401312, 'bagging_freq': 3, 'min_child_samples': 26}. Best is trial
62 with value: 0.025279323598135478.
[I 2023-12-13 10:43:14,570] Trial 66 finished with value: 0.02529983135180097
```

and parameters: {'lambda': 0.012702419621684082, 'alpha': 1.4948995533004702e-07, 'colsample_bytree': 0.9773931240309349, 'subsample': 0.9820864517247113, 'learning_rate': 0.016936037134935425, 'n_estimators': 879, 'max_depth': 28, 'bagging_fraction': 0.2705690999367246, 'feature_fraction': 0.5811281992007445, 'bagging_freq': 3, 'min_child_samples': 14}. Best is trial 62 with value: 0.025279323598135478.
[I 2023-12-13 10:43:59,507] Trial 67 finished with value: 0.025304702345009207 and parameters: {'lambda': 0.00929988999091103, 'alpha': 2.513835702769401e-08, 'colsample_bytree': 0.9351106375663955, 'subsample': 0.9133410839401552, 'learning_rate': 0.010235852209524363, 'n_estimators': 134, 'max_depth': 30, 'bagging_fraction': 0.23256717323426424, 'feature_fraction': 0.6270552590619286, 'bagging_freq': 2, 'min_child_samples': 19}. Best is trial 62 with value: 0.025279323598135478.
[I 2023-12-13 10:44:35,666] Trial 68 finished with value: 0.02528404817683199 and parameters: {'lambda': 0.0007228239388269606, 'alpha': 7.844405884424991e-08, 'colsample_bytree': 0.9732719593859215, 'subsample': 0.9447588083912639, 'learning_rate': 0.01935383654868368, 'n_estimators': 104, 'max_depth': 28, 'bagging_fraction': 0.26279550784786043, 'feature_fraction': 0.5501639325747597, 'bagging_freq': 3, 'min_child_samples': 25}. Best is trial 62 with value: 0.025279323598135478.
[I 2023-12-13 10:45:19,140] Trial 69 finished with value: 0.02530844540343384 and parameters: {'lambda': 0.0006131068157811571, 'alpha': 1.0841553305521246e-08, 'colsample_bytree': 0.8722986883488371, 'subsample': 0.9396254464112339, 'learning_rate': 0.02285773057705424, 'n_estimators': 101, 'max_depth': 29, 'bagging_fraction': 0.24996343818030248, 'feature_fraction': 0.7049542394739258, 'bagging_freq': 3, 'min_child_samples': 24}. Best is trial 62 with value: 0.025279323598135478.
[I 2023-12-13 10:46:08,696] Trial 70 finished with value: 0.025271048307844695 and parameters: {'lambda': 0.0012040381174800252, 'alpha': 6.030370361328975e-08, 'colsample_bytree': 0.9065786456016266, 'subsample': 0.982699436776618, 'learning_rate': 0.029978382217303604, 'n_estimators': 176, 'max_depth': 30, 'bagging_fraction': 0.2807204261206366, 'feature_fraction': 0.5545735374667086, 'bagging_freq': 2, 'min_child_samples': 29}. Best is trial 70 with value: 0.025271048307844695.
[I 2023-12-13 10:46:41,796] Trial 71 finished with value: 0.025329212165593355 and parameters: {'lambda': 0.0010876208771164685, 'alpha': 5.7489241918688196e-08, 'colsample_bytree': 0.907968827955973, 'subsample': 0.9945827309791717, 'learning_rate': 0.02904739322869467, 'n_estimators': 967, 'max_depth': 28, 'bagging_fraction': 0.2860412368259299, 'feature_fraction': 0.5465621469451375, 'bagging_freq': 2, 'min_child_samples': 28}. Best is trial 70 with value: 0.025271048307844695.
[I 2023-12-13 10:47:16,370] Trial 72 finished with value: 0.025360465908549713 and parameters: {'lambda': 0.00026995106532847206, 'alpha': 1.9182671560896877e-07, 'colsample_bytree': 0.9450974144979222, 'subsample': 0.96057642610671, 'learning_rate': 0.0390002727275431, 'n_estimators': 166, 'max_depth': 30, 'bagging_fraction': 0.21988418419407532, 'feature_fraction': 0.5559959274654384, 'bagging_freq': 3, 'min_child_samples': 35}. Best is trial 70 with value: 0.025271048307844695.

```
[I 2023-12-13 10:47:45,365] Trial 73 finished with value: 0.025289374951137627
and parameters: {'lambda': 0.0029642602887767285, 'alpha':
1.0728036979945211e-07, 'colsample_bytree': 0.9812408123631056, 'subsample':
0.9389773905747183, 'learning_rate': 0.01994379490960799, 'n_estimators': 662,
'max_depth': 29, 'bagging_fraction': 0.2002304071712448, 'feature_fraction':
0.6157143387214091, 'bagging_freq': 1, 'min_child_samples': 99}. Best is trial
70 with value: 0.025271048307844695.
[I 2023-12-13 10:48:20,062] Trial 74 finished with value: 0.025314575269197014
and parameters: {'lambda': 0.0017986906296417798, 'alpha':
2.6372927193020125e-08, 'colsample_bytree': 0.9296157549019183, 'subsample':
0.9788628015020875, 'learning_rate': 0.026182503688699753, 'n_estimators': 128,
'max_depth': 30, 'bagging_fraction': 0.2595975698485283, 'feature_fraction':
0.5074140356087514, 'bagging_freq': 2, 'min_child_samples': 23}. Best is trial
70 with value: 0.025271048307844695.
[I 2023-12-13 10:48:44,858] Trial 75 finished with value: 0.025294108075544247
and parameters: {'lambda': 0.00013098102455342818, 'alpha':
8.117088809436503e-08, 'colsample_bytree': 0.8976485825273015, 'subsample':
0.9048189270632825, 'learning_rate': 0.013686813276809713, 'n_estimators': 179,
'max_depth': 28, 'bagging_fraction': 0.29554872894331374, 'feature_fraction':
0.5645462205986478, 'bagging_freq': 3, 'min_child_samples': 32}. Best is trial
70 with value: 0.025271048307844695.
[I 2023-12-13 10:49:09,925] Trial 76 finished with value: 0.025296634801082026
and parameters: {'lambda': 0.0007811424846940675, 'alpha':
2.624597131307692e-07, 'colsample_bytree': 0.9618687467846792, 'subsample':
0.9493825620806484, 'learning_rate': 0.028413953883698252, 'n_estimators': 957,
'max_depth': 27, 'bagging_fraction': 0.23504613529941554, 'feature_fraction':
0.5194164642670636, 'bagging_freq': 2, 'min_child_samples': 20}. Best is trial
70 with value: 0.025271048307844695.
[I 2023-12-13 10:49:33,110] Trial 77 finished with value: 0.02526263993093133
and parameters: {'lambda': 0.0004637456474473309, 'alpha':
3.819636291959033e-08, 'colsample_bytree': 0.9995879616571358, 'subsample':
0.9194324999028596, 'learning_rate': 0.02187349504206808, 'n_estimators': 793,
'max_depth': 26, 'bagging_fraction': 0.27906915989463854, 'feature_fraction':
0.5326148862967298, 'bagging_freq': 3, 'min_child_samples': 26}. Best is trial
77 with value: 0.02526263993093133.
[I 2023-12-13 10:49:50,757] Trial 78 finished with value: 0.02526819691407862
and parameters: {'lambda': 0.0004141635053955127, 'alpha':
1.4022317039661499e-08, 'colsample_bytree': 0.9993501415842129, 'subsample':
0.8915074041959273, 'learning_rate': 0.04017417862434065, 'n_estimators': 882,
'max_depth': 24, 'bagging_fraction': 0.2826374533840479, 'feature_fraction':
0.5897895832389817, 'bagging_freq': 3, 'min_child_samples': 28}. Best is trial
77 with value: 0.02526263993093133.
[I 2023-12-13 10:50:06,958] Trial 79 finished with value: 0.025364609935982094
and parameters: {'lambda': 0.00038322596005400256, 'alpha':
2.0368701154004353e-08, 'colsample_bytree': 0.8641166470533846, 'subsample':
0.8511610339097145, 'learning_rate': 0.04064435432833737, 'n_estimators': 798,
'max_depth': 25, 'bagging_fraction': 0.2825322785402769, 'feature_fraction':
0.5842551402777211, 'bagging_freq': 3, 'min_child_samples': 37}. Best is trial
```

77 with value: 0.02526263993093133.
[I 2023-12-13 10:50:20,292] Trial 80 finished with value: 0.025272425128560195 and parameters: {'lambda': 0.0015636025995438537, 'alpha': 3.7312194727316824e-08, 'colsample_bytree': 0.9215349904278742, 'subsample': 0.8813271039788815, 'learning_rate': 0.03573699056031966, 'n_estimators': 891, 'max_depth': 22, 'bagging_fraction': 0.30414931723452127, 'feature_fraction': 0.6067729093045267, 'bagging_freq': 2, 'min_child_samples': 28}. Best is trial 77 with value: 0.02526263993093133.
[I 2023-12-13 10:50:34,572] Trial 81 finished with value: 0.02532528233324931 and parameters: {'lambda': 0.0015222172142452884, 'alpha': 3.3261206030900194e-08, 'colsample_bytree': 0.918666707261028, 'subsample': 0.9193329600816433, 'learning_rate': 0.04523114682638095, 'n_estimators': 870, 'max_depth': 22, 'bagging_fraction': 0.30145329685624617, 'feature_fraction': 0.6099933555587279, 'bagging_freq': 2, 'min_child_samples': 29}. Best is trial 77 with value: 0.02526263993093133.
[I 2023-12-13 10:50:53,507] Trial 82 finished with value: 0.025272871767711046 and parameters: {'lambda': 0.000444225496369312, 'alpha': 1.0174668660723827e-08, 'colsample_bytree': 0.9975707140908492, 'subsample': 0.8931819942897156, 'learning_rate': 0.02597222831974006, 'n_estimators': 908, 'max_depth': 24, 'bagging_fraction': 0.3340417761890419, 'feature_fraction': 0.5293789171166005, 'bagging_freq': 2, 'min_child_samples': 33}. Best is trial 77 with value: 0.02526263993093133.
[I 2023-12-13 10:51:05,642] Trial 83 finished with value: 0.025251499509979125 and parameters: {'lambda': 0.00014850645826293993, 'alpha': 1.0160594276574543e-08, 'colsample_bytree': 0.9990053643429847, 'subsample': 0.9056310863413776, 'learning_rate': 0.03602661957810861, 'n_estimators': 903, 'max_depth': 21, 'bagging_fraction': 0.2501314865582333, 'feature_fraction': 0.5971070725851176, 'bagging_freq': 1, 'min_child_samples': 33}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:51:16,778] Trial 84 finished with value: 0.0253302679627646 and parameters: {'lambda': 0.00021885099543564408, 'alpha': 1.0969923900132033e-08, 'colsample_bytree': 0.9978199765750132, 'subsample': 0.8669085313336257, 'learning_rate': 0.036079059577199146, 'n_estimators': 906, 'max_depth': 21, 'bagging_fraction': 0.25240189346055997, 'feature_fraction': 0.5304537676743629, 'bagging_freq': 1, 'min_child_samples': 32}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:51:35,250] Trial 85 finished with value: 0.025414583946372277 and parameters: {'lambda': 0.0003518669179381192, 'alpha': 1.5931274431388616e-08, 'colsample_bytree': 0.9491359197665606, 'subsample': 0.8947690749270222, 'learning_rate': 0.05142816205451055, 'n_estimators': 846, 'max_depth': 24, 'bagging_fraction': 0.3324207773174931, 'feature_fraction': 0.4930862663131579, 'bagging_freq': 1, 'min_child_samples': 37}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:51:46,822] Trial 86 finished with value: 0.025293114862564223 and parameters: {'lambda': 0.00013268426023313015, 'alpha': 4.07209074379293e-08, 'colsample_bytree': 0.9495688710924934, 'subsample': 0.8793808175776866, 'learning_rate': 0.03689995387346901, 'n_estimators': 999, 'max_depth': 21, 'bagging_fraction': 0.2845057472752774, 'feature_fraction':

0.5365805565704731, 'bagging_freq': 1, 'min_child_samples': 33}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:52:00,782] Trial 87 finished with value: 0.025312845591834332 and parameters: {'lambda': 5.6802699916679176e-05, 'alpha': 1.8019098999611515e-08, 'colsample_bytree': 0.9927243593544947, 'subsample': 0.8322740983671587, 'learning_rate': 0.030161447035465812, 'n_estimators': 892, 'max_depth': 23, 'bagging_fraction': 0.31515715307890263, 'feature_fraction': 0.5688657876954213, 'bagging_freq': 2, 'min_child_samples': 28}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:52:17,392] Trial 88 finished with value: 0.02528253270577153 and parameters: {'lambda': 0.00038893823154504314, 'alpha': 4.401412895793435e-08, 'colsample_bytree': 0.9999460771157064, 'subsample': 0.8601133432690624, 'learning_rate': 0.023659583210202825, 'n_estimators': 936, 'max_depth': 24, 'bagging_fraction': 0.24266686603428936, 'feature_fraction': 0.5962318455754821, 'bagging_freq': 2, 'min_child_samples': 43}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:52:41,344] Trial 89 finished with value: 0.025332818859468705 and parameters: {'lambda': 0.0007813079705843958, 'alpha': 1.0718604368760781e-08, 'colsample_bytree': 0.9306090551703622, 'subsample': 0.9129751544670698, 'learning_rate': 0.04603844014434903, 'n_estimators': 809, 'max_depth': 26, 'bagging_fraction': 0.28022545193815496, 'feature_fraction': 0.5331058274997771, 'bagging_freq': 1, 'min_child_samples': 26}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:53:00,999] Trial 90 finished with value: 0.025355172221162202 and parameters: {'lambda': 0.0004542094635063299, 'alpha': 2.4771205393281695e-08, 'colsample_bytree': 0.8995266344487026, 'subsample': 0.8896997607181751, 'learning_rate': 0.04222904737319777, 'n_estimators': 755, 'max_depth': 25, 'bagging_fraction': 0.3077489672983512, 'feature_fraction': 0.5592734390273437, 'bagging_freq': 0, 'min_child_samples': 29}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:53:18,345] Trial 91 finished with value: 0.025279997593325304 and parameters: {'lambda': 0.0002027018425549025, 'alpha': 4.753226504955371e-08, 'colsample_bytree': 0.9856794922995038, 'subsample': 0.8540732795725582, 'learning_rate': 0.023816883653017176, 'n_estimators': 943, 'max_depth': 24, 'bagging_fraction': 0.24515034831314528, 'feature_fraction': 0.6018377243248143, 'bagging_freq': 2, 'min_child_samples': 38}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:53:32,768] Trial 92 finished with value: 0.02527458836912782 and parameters: {'lambda': 0.00020584827962403675, 'alpha': 6.246210299665723e-08, 'colsample_bytree': 0.9573321335766642, 'subsample': 0.926117788395477, 'learning_rate': 0.03290499817562831, 'n_estimators': 907, 'max_depth': 22, 'bagging_fraction': 0.2597134148248516, 'feature_fraction': 0.6013824879826534, 'bagging_freq': 2, 'min_child_samples': 35}. Best is trial 83 with value: 0.025251499509979125.
[I 2023-12-13 10:53:44,300] Trial 93 finished with value: 0.02530303504336863 and parameters: {'lambda': 0.000195024068158722, 'alpha': 4.2391068606964633e-08, 'colsample_bytree': 0.9578544092431202, 'subsample': 0.9275062977580407, 'learning_rate': 0.03293392229800676, 'n_estimators': 902,

```
'max_depth': 21, 'bagging_fraction': 0.2617174426402094, 'feature_fraction':
0.6095496329618074, 'bagging_freq': 2, 'min_child_samples': 34}. Best is trial
83 with value: 0.025251499509979125.
[I 2023-12-13 10:53:52,882] Trial 94 finished with value: 0.02530267928259682
and parameters: {'lambda': 8.680957103528491e-05, 'alpha':
1.7986329387478453e-08, 'colsample_bytree': 0.988173117448708, 'subsample':
0.8956250569135278, 'learning_rate': 0.024371342813831604, 'n_estimators': 962,
'max_depth': 19, 'bagging_fraction': 0.23228304305730701, 'feature_fraction':
0.6479164772855127, 'bagging_freq': 2, 'min_child_samples': 38}. Best is trial
83 with value: 0.025251499509979125.
[I 2023-12-13 10:54:06,453] Trial 95 finished with value: 0.025279717873417228
and parameters: {'lambda': 0.0005744235061457311, 'alpha':
7.533081928234449e-08, 'colsample_bytree': 0.9817598620534969, 'subsample':
0.8478958204152356, 'learning_rate': 0.03643284819410564, 'n_estimators': 860,
'max_depth': 23, 'bagging_fraction': 0.33668097989370943, 'feature_fraction':
0.5687844647884539, 'bagging_freq': 1, 'min_child_samples': 45}. Best is trial
83 with value: 0.025251499509979125.
[I 2023-12-13 10:54:18,134] Trial 96 finished with value: 0.0253112393980366 and
parameters: {'lambda': 0.0015509418169606943, 'alpha': 7.832882759871663e-08,
'colsample_bytree': 0.9521428143612222, 'subsample': 0.8697610998571947,
'learning_rate': 0.03905284849226079, 'n_estimators': 861, 'max_depth': 21,
'bagging_fraction': 0.32788833577203863, 'feature_fraction': 0.5685443770627205,
'bagging_freq': 1, 'min_child_samples': 34}. Best is trial 83 with value:
0.025251499509979125.
[I 2023-12-13 10:54:30,443] Trial 97 finished with value: 0.02534378532917596
and parameters: {'lambda': 0.00047170044102644244, 'alpha':
1.0064417887817642e-08, 'colsample_bytree': 0.9255372537060048, 'subsample':
0.8279544456649595, 'learning_rate': 0.03422443374736416, 'n_estimators': 835,
'max_depth': 22, 'bagging_fraction': 0.33986519823847217, 'feature_fraction':
0.5277712785473855, 'bagging_freq': 1, 'min_child_samples': 49}. Best is trial
83 with value: 0.025251499509979125.
[I 2023-12-13 10:54:46,226] Trial 98 finished with value: 0.025291617682891294
and parameters: {'lambda': 0.0009944506260975454, 'alpha':
2.7701021622764098e-08, 'colsample_bytree': 0.9620560778194718, 'subsample':
0.9363000954113884, 'learning_rate': 0.028293047283363543, 'n_estimators': 782,
'max_depth': 23, 'bagging_fraction': 0.2758468239749495, 'feature_fraction':
0.6182969254758885, 'bagging_freq': 2, 'min_child_samples': 45}. Best is trial
83 with value: 0.025251499509979125.
[I 2023-12-13 10:55:01,185] Trial 99 finished with value: 0.025338250072378693
and parameters: {'lambda': 0.002136646904040068, 'alpha': 6.421205769671231e-08,
'colsample_bytree': 0.9411428359995448, 'subsample': 0.9866519069445011,
'learning_rate': 0.04743222155107119, 'n_estimators': 913, 'max_depth': 22,
'bagging_fraction': 0.31143060488478047, 'feature_fraction': 0.6525176227963684,
'bagging_freq': 1, 'min_child_samples': 30}. Best is trial 83 with value:
0.025251499509979125.
```

```
[ ]: def cal_ICIR(data: pd.DataFrame, feild: str) -> tuple[float, float]:
         """
         data is a dataframe with columns: date, return, factor feild
         feild is the factor name
         return IC and IR
         """
         data = data.loc[:, ['date', 'return', feild]]
         data.dropna(inplace=True)
         IC_dataframe = data.groupby('date').apply(lambda x: x.
       ↪corr(method='spearman')[feild]['return'])
         return IC_dataframe.mean(), IC_dataframe.mean()/IC_dataframe.std()

     def test_factor(ICIR: tuple[float, float]) -> str:
         """
         ICIR is a tuple of IC and IR
         return the test result
         """
         if abs(ICIR[0]) > 0.01 and abs(ICIR[1]) > 0.03:
             return 'pass'
         else:
             return 'fail'
```

```
[ ]: import joblib
     model = joblib.load('./model/lightGBM.pkl')
```

```
[ ]: dtrain = lgb.Dataset(X, label=y)
     model = lgb.train(best_params_lgb, dtrain)
     preds = model.predict(X_outsample_test)
     matrix = pd.concat([other_info_outsample_test, pd.DataFrame(preds,␣
       ↪columns=['preds'], index = X_outsample_test.index), y_outsample_test],␣
       ↪axis=1)
     ICIR = cal_ICIR(matrix, "preds")
     print(ICIR)
     result = test_factor(ICIR)
```

```
(0.04098354371529285, 0.10894446726011417)
```

```
[ ]: import joblib
     joblib.dump(model, filename='./model/lightGBM.pkl')
```

```
[ ]: ['./model/lightGBM.pkl']
```

```
[ ]: matrix.to_csv('./result/lightGBM.csv')
```

```
[ ]: dtrain = xgb.DMatrix(X, label=y)
     model = xgb.train(best_params_xgb, dtrain)
     preds = model.predict(xgb.DMatrix(X_outsample_test))
```

```
matrix = pd.concat([other_info_outsample_test, pd.DataFrame(preds,␣
 ↪columns=['preds'], index = X_outsample_test.index), y_outsample_test],␣
 ↪axis=1)
ICIR = cal_ICIR(matrix, "preds")
print(ICIR)
result = test_factor(ICIR)
print(result)
```

```
(0.03939948351261069, 0.10843730464715315)
pass
```

[ ]:
```
import joblib
joblib.dump(model, filename='./model/XGBoost.pkl')
```

[ ]: ['./model/XGBoost.pkl']

[ ]:
```
matrix.to_csv('./result/XGBoost.csv')
```

[ ]:
```
abs_IC = []
abs_IR = []
for i in outdsample_data.columns[4:-1]:
    ICIR = cal_ICIR(outdsample_data, i)
    abs_IC.append(abs(ICIR[0]))
    abs_IR.append(abs(ICIR[1]))
    print(i)
    print(ICIR)
    result = test_factor(ICIR)
    print(result)
```

```
dividendyield
(0.015824726607574644, 0.042184647306419805)
pass
day_momentum
(-0.03689649790894963, -0.09217629437455241)
pass
val_ortoev_ttm
(0.004996381701627184, 0.010633179127117202)
fail
val_lnmv
(-0.010571168040773474, -0.02969597672129529)
fail
val_lntotassets
(0.0010502788330560924, 0.003110385890002309)
fail
fa_sellexpensetogr_ttm
(-0.003763208351110012, -0.010505331561647416)
fail
fa_salestocost_ttm
```

```
(0.0028314831635785215, 0.007291340941836725)
fail
mmt_discret_w
(-0.012501906316337128, -0.034048293924522015)
pass
vol_up_std_m
(-0.001158797691518292, -0.002693989094036354)
fail
dividendyield2
(0.004289274651796401, 0.009095240775840676)
fail
val_floatmv
(-0.0024735025088819806, -0.0075502323326710075)
fail
risk_variance20
(-0.009785025227259643, -0.02135754289821015)
fail
risk_lossvariance20
(-0.017069249920817615, -0.038097589977360576)
pass
risk_beta20
(0.010234820763753061, 0.024995799227719252)
fail
risk_volatilityratio
(-0.013005732597344619, -0.029162272673061564)
fail
risk_hisrelation
(-0.001441978074287302, -0.0035244772111479995)
fail
risk_revsvarratio
(-0.012481426753688983, -0.029578191177139307)
fail
fa_operincometopbt
(-0.009503046040429404, -0.026925380060587138)
fail
fa_octogr_ttm
(0.019731197614579584, 0.0543580827941368)
pass
fa_netprofitmargin_ttm
(-0.01710687269746506, -0.04825639989170216)
pass
fa_salescashtoor
(-0.007915464923685743, -0.017652525366289463)
fail
fa_cashrecovratio_ttm
(0.026512998470046022, 0.07745444774194984)
pass
fa_blev
```

```
(-0.018572959184153297, -0.05645075343469298)
pass
fa_current
(-0.007320708279582767, -0.022212844881620723)
fail
fa_apturn_ttm
(-0.004758114551516592, -0.013341822957780526)
fail
fa_ncgr_ttm
(0.0004440169096252624, 0.0013220020126310454)
fail
continuous_count
(nan, nan)
fail
```

[ ]:
```python
sorted(abs_IC, reverse=True)
```

[ ]:
```
[0.03689649790894963,
 0.026512998470046022,
 0.019731197614579584,
 0.018572959184153297,
 0.01710687269746506,
 0.017069249920817615,
 0.015824726607574644,
 0.013005732597344619,
 0.012501906316337128,
 0.012481426753688983,
 0.010571168040773474,
 0.010234820763753061,
 0.009785025227259643,
 0.009503046040429404,
 0.007915464923685743,
 0.007320708279582767,
 0.004996381701627184,
 0.004758114551516592,
 0.004289274651796401,
 0.003763208351110012,
 0.0028314831635785215,
 0.0024735025088819806,
 0.001441978074287302,
 0.001158797691518292,
 0.0010502788330560924,
 0.0004440169096252624,
 nan]
```

[ ]:
```python
sorted(abs_IR, reverse=True)
```

```
[ ]: [0.09217629437455241,
      0.07745444774194984,
      0.05645075343469298,
      0.0543580827941368,
      0.04825639989170216,
      0.042184647306419805,
      0.038097589977360576,
      0.034048293924522015,
      0.02969597672129529,
      0.029578191177139307,
      0.029162272673061564,
      0.026925380060587138,
      0.024995799227719252,
      0.022212844881620723,
      0.02135754289821015,
      0.017652525366289463,
      0.013341822957780526,
      0.010633179127117202,
      0.010505331561647416,
      0.009095240775840676,
      0.0075502323326710075,
      0.007291340941836725,
      0.0035244772111479995,
      0.003110385890002309,
      0.002693989094036354,
      0.0013220020126310454,
      nan]
```

Portfolio

December 17, 2023

[ ]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cvxpy as cp
import joblib
import xgboost as xgb
```

```
(CVXPY) Dec 17 10:13:02 PM: Encountered unexpected exception importing solver
OSQP:
ImportError('DLL load failed while importing qdldl:      ')
```

[ ]:
```python
data = pd.read_csv('factor_pass.csv', index_col=0)
data.fillna(0, inplace=True)
```

[ ]:
```python
data
```

[ ]:
```
            id  name        date     evebit  dividendyield  day_momentum  \
0          338      2016-01-04  12.680916       3.893215      0.000000
1          338      2016-01-05  12.680916       3.941441      0.000000
2          338      2016-01-06  12.680916       3.804348      0.000000
3          338      2016-01-07  12.680916       4.171633      0.000000
4          338      2016-01-08  12.680916       4.046243      0.000000
…          …    …         …          …             …            …
16974  601689      2022-12-26  66.745010       0.746774      0.985613
16975  601689      2022-12-27  66.745010       0.747739      0.986239
16976  601689      2022-12-28  66.745010       0.795396      0.934050
16977  601689      2022-12-29  66.745010       0.790372      0.947643
16978  601689      2022-12-30  66.745010       0.790372      0.953016

       val_ortoev_ttm   val_lnmv  val_lntotassets  fa_sellexpensetogr_ttm  \
0            2.537171  24.305300          25.5089                  8.1856
1            2.563779  24.292999          25.5089                  8.1856
2            2.493336  24.328400          25.5089                  8.1856
3            2.711453  24.236300          25.5089                  8.1856
4            2.643240  24.266800          25.5089                  8.1856
…                 …          …              …                     …
16974        0.207402  24.947599          23.9587                  1.4537
16975        0.207659  24.946301          23.9587                  1.4537
```

```
16976        0.220349  24.884501          23.9587                  1.4537
16977        0.219014  24.890800          23.9587                  1.4537
16978        0.219014  24.890800          23.9587                  1.4537

       …  fa_operincometopbt  fa_octogr_ttm  fa_netprofitmargin_ttm  \
0      …              0.8770        95.4396                  3.7767
1      …              0.8770        95.4396                  3.7767
2      …              0.8770        95.4396                  3.7767
3      …              0.8770        95.4396                  3.7767
4      …              0.8770        95.4396                  3.7767
…    …                   …            …                      …
16974  …              0.9554        89.6781                  9.5286
16975  …              0.9554        89.6781                  9.5286
16976  …              0.9554        89.6781                  9.5286
16977  …              0.9554        89.6781                  9.5286
16978  …              0.9554        89.6781                  9.5286

       fa_salescashtoor  fa_cashrecovratio_ttm  fa_blev  fa_current  \
0                1.0049              10.429103   0.7290      1.3775
1                1.0049              10.429103   0.7290      1.3775
2                1.0049              10.429103   0.7290      1.3775
3                1.0049              10.429103   0.7290      1.3775
4                1.0049              10.429103   0.7290      1.3775
…                   …                    …        …          …
16974            1.0952               7.318087   0.2216      1.2370
16975            1.0952               7.318087   0.2216      1.2370
16976            1.0952               7.318087   0.2216      1.2370
16977            1.0952               7.318087   0.2216      1.2370
16978            1.0952               7.318087   0.2216      1.2370

       fa_apturn_ttm  fa_ncgr_ttm    return
0             2.4120     -37.4817 -0.012236
1             2.4120     -37.4817  0.036036
2             2.4120     -37.4817 -0.088043
3             2.4120     -37.4817  0.030989
4             2.4120     -37.4817 -0.036994
…                …            …        …
16974         2.1767     -76.6964 -0.001290
16975         2.1767     -76.6964 -0.059916
16976         2.1767     -76.6964  0.006356
16977         2.1767     -76.6964  0.000000
16978         2.1767     -76.6964  0.000000

[16979 rows x 31 columns]
```

[ ]:
```
returns = data.pivot_table(index='date', columns='id', values='return')
```

[ ]:
```
returns.fillna(0, inplace=True)
```

[ ]:
```
returns
```

[ ]:
```
id                338       625       951      2594    600104    600418  \
date
2016-01-04 -0.012236  0.011688  0.017576  0.010041  0.010644  0.010355
2016-01-05  0.036036  0.037227  0.020846  0.018029  0.012036  0.010981
2016-01-06 -0.088043 -0.078589 -0.093932 -0.082920 -0.059465 -0.096307
2016-01-07  0.030989  0.051041  0.028332  0.072550  0.027924  0.027244
2016-01-08 -0.036994 -0.072843 -0.058234 -0.026081 -0.026140 -0.094384
…                …         …         …         …         …         …
2022-12-26  0.007775  0.002297 -0.004032 -0.001143  0.003399  0.020573
2022-12-27 -0.021215 -0.055768 -0.028340 -0.028613 -0.008808 -0.043197
2022-12-28 -0.001970 -0.005663  0.046528  0.000550 -0.020506 -0.012039
2022-12-29  0.004936  0.001627 -0.015262  0.008675  0.005583  0.000000
2022-12-30  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000

id             600660    601238    601633    601689
date
2016-01-04 -0.002776  0.038821  0.022789 -0.057040
2016-01-05  0.025052  0.051561  0.017825  0.026227
2016-01-06 -0.059063 -0.080972 -0.080560 -0.100165
2016-01-07  0.054834  0.079295  0.016190 -0.027485
2016-01-08 -0.033516 -0.064399 -0.068416 -0.099859
…                …         …         …         …
2022-12-26  0.006803  0.002641  0.005657 -0.001290
2022-12-27 -0.013795 -0.021071 -0.033091 -0.059916
2022-12-28 -0.000285 -0.017040  0.002738  0.006356
2022-12-29  0.001428  0.006387  0.010922  0.000000
2022-12-30  0.000000  0.000000  0.000000  0.000000

[1703 rows x 10 columns]
```

[ ]:
```
def cal_turnover(weights):
    return np.sum(np.abs(weights[:-1] - weights[1:]), axis=1)
```

1 Mean Variance Framework

[ ]:
```
def cal_covariance_matrix(returns, pre_day, lambda_value = 0.94):
    """
    param:
        returns: a dataframe of returns
        pre_day: the predict day
        lambda_value: the weight of exponential decay
    return: a covariance matrix
    """
```

```
    returns = returns.loc[returns.index < pre_day, :].values
    mean_returns = np.mean(returns, axis=1)
    returns = returns - mean_returns.reshape(-1, 1)

    n = len(returns)
    weights = np.power(lambda_value, np.arange(n)[::-1])
    weighted_returns = returns * weights.reshape(-1, 1)
    exp_cov_matrix = weighted_returns.T @ returns / np.sum(weights)

    return exp_cov_matrix
```

[ ]:
```
def mpt_optimization(cov_matrix, expected_return, risk_aversion):
    """
    param:
        cov_matrix: a covariance matrix
        expected_return: a vector of expected returns
        risk_aversion: a risk aversion coefficient
    return: a vector of optimal weights
    """
    num_assets = len(expected_return)
    weights = cp.Variable(num_assets)
    objective = cp.Maximize(expected_return @ weights - 1/2 * risk_aversion *␣
 ↪cp.quad_form(weights, cov_matrix))
    constraints = [cp.sum(weights) == 1, weights >= 0]
    problem = cp.Problem(objective, constraints)
    problem.solve()
    optimal_weights = weights.value

    return optimal_weights
```

[ ]:
```
xgb_model = joblib.load("XGBoost.pkl")
```

[ ]:
```
lgb_model = joblib.load("lightGBM.pkl")
```

[ ]:
```
data['continuous_count'] = data.groupby((data['evebit'] != data['evebit'].
 ↪shift(1)).cumsum()).cumcount() + 1
data = data.reindex(columns=data.columns.tolist()[:-2] + ['continuous_count',␣
 ↪'return'])
data.dropna(inplace=True)
date_list = data.loc[data.loc[:, "date"] >= "2022-01-01", "date"].unique()
y_outsample_test = returns[returns.index >= "2022-01-01"].values
# preds = xgb_model.predict(xgb.DMatrix(data.loc[data.loc[:, "date"] ==␣
 ↪date_list[0]].drop(columns=["date", "id", "name", "return"])))
```

[ ]:
```
outsample_return_equal = np.multiply(y_outsample_test, (np.ones(len(date_list))␣
 ↪/ returns.shape[1]).reshape(-1, 1)).sum(axis=1)
```

[ ]:
```
def MPT_xgb(risk_aversion):
    weights_list = []
    for i in date_list:
        preds1 = xgb_model.predict(xgb.DMatrix(data.loc[data.loc[:, "date"] ==␣
 ↪i].drop(columns=["date", "id", "name", "return"])))
        # preds2 = lgb_model.predict(data.loc[data.loc[:, "date"] == i].
 ↪drop(columns=["date", "id", "name", "return"]))
        weights_list.append(mpt_optimization(cal_covariance_matrix(returns, i),␣
 ↪preds1, risk_aversion))
    result = np.stack(weights_list, axis=0)
    return np.multiply(y_outsample_test, result).sum(axis=1),␣
 ↪cal_turnover(result).mean()

def MPT_lgb(risk_aversion):
    weights_list = []
    for i in date_list:
        # preds1 = xgb_model.predict(xgb.DMatrix(data.loc[data.loc[:, "date"]␣
 ↪== i].drop(columns=["date", "id", "name", "return"])))
        preds2 = lgb_model.predict(data.loc[data.loc[:, "date"] == i].
 ↪drop(columns=["date", "id", "name", "return"]))
        weights_list.append(mpt_optimization(cal_covariance_matrix(returns, i),␣
 ↪preds2, risk_aversion))
    result = np.stack(weights_list, axis=0)
    return np.multiply(y_outsample_test, result).sum(axis=1),␣
 ↪cal_turnover(result).mean()
```

2 Black-Litterman

[ ]:
```
cap = pd.read_csv("TRD_Dalyr.csv")
```

[ ]:
```
grouped_cap = cap.pivot_table(index='Trddt', columns='Stkcd', values='Dsmvtll')
```

[ ]:
```
grouped_cap
```

[ ]:
```
Stkcd               338           625          951           2594    \
Trddt
2022-01-04  1.200004e+08  9.802465e+07  19596820.92  4.913617e+08
2022-01-05  1.211536e+08  9.624122e+07  18985888.86  4.591966e+08
2022-01-06  1.215606e+08  9.280176e+07  19303103.58  4.533220e+08
2022-01-07  1.204753e+08  9.216482e+07  20231250.38  4.469397e+08
2022-01-10  1.225103e+08  9.241960e+07  20430978.17  4.465771e+08
…                    …             …            …             …
2022-12-26  6.980239e+07  1.081392e+08  17482056.08  4.758049e+08
2022-12-27  7.034507e+07  1.083876e+08  17411563.92  4.752610e+08
2022-12-28  6.885270e+07  1.023431e+08  16918118.78  4.616624e+08
2022-12-29  6.871703e+07  1.017635e+08  17705281.26  4.619163e+08
```

```
2022-12-30  6.905620e+07  1.019291e+08  17435061.30  4.659233e+08

Stkcd            600104       600418       600660        601238  \
Trddt
2022-01-04  2.417308e+08  36691364.49  93940058.97  1.111157e+08
2022-01-05  2.447685e+08  34026872.54  94340656.24  1.092977e+08
2022-01-06  2.407961e+08  34179753.23  92217490.73  1.066071e+08
2022-01-07  2.413803e+08  33611910.68  94080268.01  1.057345e+08
2022-01-10  2.424318e+08  32847507.26  96744239.84  1.063162e+08
…                    …            …            …             …
2022-12-26  1.718637e+08  29724373.26  70665357.79  8.367023e+07
2022-12-27  1.724479e+08  30335896.00  71146074.51  8.389119e+07
2022-12-28  1.709290e+08  29025490.12  70164611.21  8.212351e+07
2022-12-29  1.674240e+08  28676048.56  70144581.35  8.072410e+07
2022-12-30  1.683587e+08  28676048.56  70244730.66  8.123967e+07

Stkcd            601633       601689
Trddt
2022-01-04  2.931864e+08  56534989.14
2022-01-05  2.821412e+08  55543147.23
2022-01-06  2.737347e+08  55014164.87
2022-01-07  2.771709e+08  53989261.56
2022-01-10  2.877865e+08  57460708.26
…                    …            …
2022-12-26  1.853609e+08  68326887.46
2022-12-27  1.864095e+08  68238723.74
2022-12-28  1.802411e+08  64150130.96
2022-12-29  1.807346e+08  64557888.19
2022-12-30  1.827085e+08  64557888.19

[242 rows x 10 columns]
```

[ ]:
```python
grouped_cap = grouped_cap.reindex(columns = returns.columns)
```

[ ]:
```python
grouped_cap = grouped_cap/grouped_cap.sum(axis = 0)
```

[ ]:
```python
grouped_cap
```

[ ]:
```
id              338       625       951      2594    600104    600418  \
Trddt
2022-01-04  0.005972  0.003912  0.005381  0.004084  0.005172  0.004942
2022-01-05  0.006029  0.003841  0.005213  0.003816  0.005237  0.004583
2022-01-06  0.006049  0.003704  0.005300  0.003768  0.005152  0.004603
2022-01-07  0.005995  0.003678  0.005555  0.003715  0.005165  0.004527
2022-01-10  0.006097  0.003689  0.005610  0.003712  0.005187  0.004424
…                …         …         …         …         …         …
2022-12-26  0.003474  0.004316  0.004800  0.003955  0.003677  0.004003
```

```
2022-12-27  0.003501  0.004326  0.004781  0.003950  0.003690  0.004086
2022-12-28  0.003426  0.004085  0.004645  0.003837  0.003657  0.003909
2022-12-29  0.003420  0.004062  0.004861  0.003839  0.003582  0.003862
2022-12-30  0.003437  0.004068  0.004787  0.003872  0.003602  0.003862

id            600660    601238    601633    601689
Trddt
2022-01-04  0.004927  0.004664  0.006031  0.003231
2022-01-05  0.004948  0.004587  0.005804  0.003174
2022-01-06  0.004837  0.004474  0.005631  0.003144
2022-01-07  0.004935  0.004438  0.005702  0.003085
2022-01-10  0.005074  0.004462  0.005920  0.003284
…                …         …         …         …
2022-12-26  0.003706  0.003512  0.003813  0.003905
2022-12-27  0.003732  0.003521  0.003835  0.003899
2022-12-28  0.003680  0.003447  0.003708  0.003666
2022-12-29  0.003679  0.003388  0.003718  0.003689
2022-12-30  0.003684  0.003410  0.003759  0.003689

[242 rows x 10 columns]
```

[ ]:
```python
def black_litterman_cvxpy(expected_return, cov_matrix, view_matrix, risk_aversion, tau=0.1):
    num_assets = len(expected_return)
    P = view_matrix[:, :-1]
    Q = view_matrix[:, -1]
    omega = tau * P @ cov_matrix @ P.T
    omega = np.diag(np.diag(omega))
    cov_pinv = np.linalg.pinv(cov_matrix)
    inverse_covariance = np.linalg.pinv(np.linalg.pinv(tau * cov_pinv) + P.T @ np.linalg.pinv(omega) @ P)
    pi_adj = inverse_covariance @ (np.linalg.pinv(tau * cov_pinv) @ expected_return + P.T @ np.linalg.pinv(omega) @ Q)
    weights = cp.Variable(num_assets)
    portfolio_expected_return = cp.sum(weights @ pi_adj)
    objective = cp.Maximize(portfolio_expected_return - 1/2 * risk_aversion * cp.quad_form(weights, cov_matrix))

    constraints = [cp.sum(weights) == 1, weights >= 0]

    problem = cp.Problem(objective, constraints)

    problem.solve()

    return weights.value
```

[ ]:
```python
def LB_xgb(risk_aversion):
    weights_list = []

    for i in date_list:
        preds = xgb_model.predict(xgb.DMatrix(data.loc[data.loc[:, "date"] ==␣
↪i].drop(columns=["date", "id", "name", "return"])))
        cov_matrix = cal_covariance_matrix(returns, i)
        market_expected_return = risk_aversion * cov_matrix @ grouped_cap.
↪loc[i].values.reshape(-1, 1)
        view_matrix = np.vstack((np.eye(len(preds)), preds)).T
        weights_list.append(black_litterman_cvxpy(market_expected_return,␣
↪cov_matrix, view_matrix, risk_aversion))

    result = np.stack(weights_list, axis=0)
    return np.multiply(y_outsample_test, result).sum(axis=1),␣
↪cal_turnover(result).mean()

def LB_lgb(risk_aversion):
    weights_list = []

    for i in date_list:
        preds = lgb_model.predict(data.loc[data.loc[:, "date"] == i].
↪drop(columns=["date", "id", "name", "return"]))
        cov_matrix = cal_covariance_matrix(returns, i)
        market_expected_return = risk_aversion * cov_matrix @ grouped_cap.
↪loc[i].values.reshape(-1, 1)
        view_matrix = np.vstack((np.eye(len(preds)), preds)).T
        weights_list.append(black_litterman_cvxpy(market_expected_return,␣
↪cov_matrix, view_matrix, risk_aversion))

    result = np.stack(weights_list, axis=0)
    return np.multiply(y_outsample_test, result).sum(axis=1),␣
↪cal_turnover(result).mean()
```

[ ]:
```python
exceeded_return_lgb = []
exceeded_return_xgb = []
exceeded_sharpe_lgb = []
exceeded_sharpe_xgb = []
turnover_MPT_lgb = []
turnover_MPT_xgb = []
turnover_BL_lgb = []
turnover_BL_xgb = []

for gamma in [0.1, 0.5, 1, 2, 5, 10]:
    outsample_return_BL_lgb, turnover_BL_lgb_ = LB_lgb(gamma)
    outsample_return_BL_xgb, turnover_BL_xgb_ = LB_xgb(gamma)
    outsample_return_MPT_lgb, turnover_MPT_lgb_ = MPT_lgb(gamma)
```

```
    outsample_return_MPT_xgb, turnover_MPT_xgb_ = MPT_xgb(gamma)
    turnover_BL_lgb.append(turnover_BL_lgb_)
    turnover_BL_xgb.append(turnover_BL_xgb_)
    turnover_MPT_lgb.append(turnover_MPT_lgb_)
    turnover_MPT_xgb.append(turnover_MPT_xgb_)
    realized_return_BL_lgb =  np.cumprod(outsample_return_BL_lgb + 1) - 1
    realized_return_BL_xgb =  np.cumprod(outsample_return_BL_xgb + 1) - 1
    realized_return_MPT_lgb =  np.cumprod(outsample_return_MPT_lgb + 1) - 1
    realized_return_MPT_xgb =  np.cumprod(outsample_return_MPT_xgb + 1) - 1
    exceeded_return_lgb.append((realized_return_BL_lgb -␣
↪realized_return_MPT_lgb)[-1])
    exceeded_return_xgb.append((realized_return_BL_xgb -␣
↪realized_return_MPT_xgb)[-1])
    exceeded_sharpe_lgb.append(np.mean(outsample_return_BL_lgb -␣
↪outsample_return_MPT_lgb)/np.std(outsample_return_BL_lgb -␣
↪outsample_return_MPT_lgb))
    exceeded_sharpe_xgb.append(np.mean(outsample_return_BL_xgb -␣
↪outsample_return_MPT_xgb)/np.std(outsample_return_BL_xgb -␣
↪outsample_return_MPT_xgb))
```

[ ]:
```
exceeded_return_lgb
```

[ ]:
```
[0.007824805285031822,
 0.01617469387964421,
 0.004625431497750188,
 -0.004789655493399603,
 -0.004013228340480346,
 0.02913280547074626]
```

[ ]:
```
exceeded_return_xgb
```

[ ]:
```
[-0.5104283283521001,
 -0.577929459201749,
 -0.5956545062086224,
 -0.5740488871392581,
 -0.36210465920241974,
 -0.14569607764274828]
```

[ ]:
```
exceeded_sharpe_lgb
```

[ ]:
```
[0.010354294287765619,
 0.011143876499892132,
 0.009330195195774521,
 0.008196720403664984,
 0.00883749931604193,
 0.014861508571024432]
```

[ ]:
```
exceeded_sharpe_xgb
```

[ ]:
```
[-0.03621261780211595,
 -0.04228173173918555,
 -0.04445714976740577,
 -0.04345611943335083,
 -0.025792783566352378,
 -0.0037645975348936512]
```

[ ]:
```
turnover_BL_lgb
```

[ ]:
```
[0.32853441129936034,
 0.3393416213765903,
 0.3463704330662345,
 0.3567726347306411,
 0.38577037919244983,
 0.41547019011228076]
```

[ ]:
```
turnover_BL_xgb
```

[ ]:
```
[0.1410788485328268,
 0.14484384057535024,
 0.1496313703411349,
 0.16012760261745607,
 0.18400930976186264,
 0.2127107671343873]
```

[ ]:
```
turnover_MPT_lgb
```

[ ]:
```
[1.2833202015360017,
 1.2805866164854007,
 1.2820331152340347,
 1.274594434604934,
 1.2568204355624302,
 1.2078580849076554]
```

[ ]:
```
turnover_MPT_xgb
```

[ ]:
```
[0.9163181614904276,
 0.9303533337947273,
 0.9335859953934073,
 0.9298628727396585,
 0.8856115954984538,
 0.8403864683607816]
```

[ ]:
```
exceeded_turnover_lgb = np.array(turnover_BL_lgb) - np.array(turnover_MPT_lgb)
exceeded_turnover_xgb = np.array(turnover_BL_xgb) - np.array(turnover_MPT_xgb)
```

[ ]:
```
exceeded_turnover_lgb
```

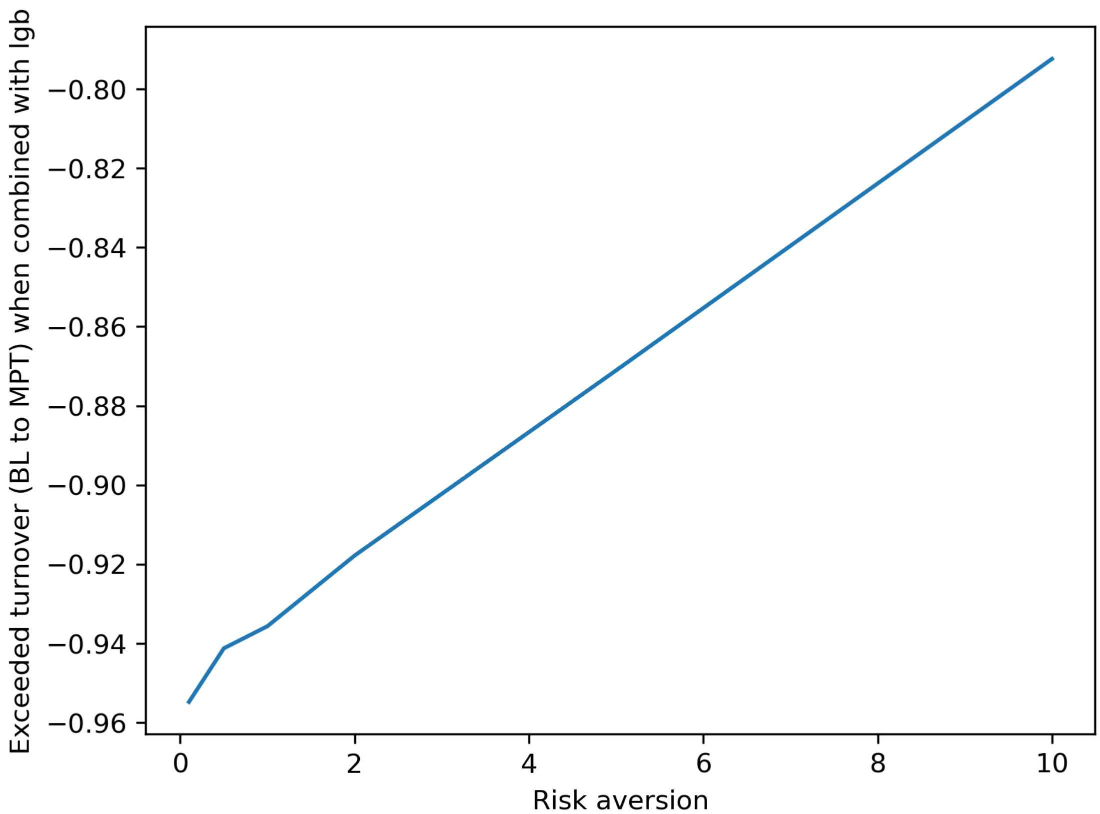
[ ]:
```
array([-0.95478579, -0.941245  , -0.93566268, -0.9178218 , -0.87105006,
       -0.79238789])
```

[ ]:
```
exceeded_turnover_xgb
```

[ ]:
```
array([-0.77523931, -0.78550949, -0.78395463, -0.76973527, -0.70160229,
       -0.6276757 ])
```

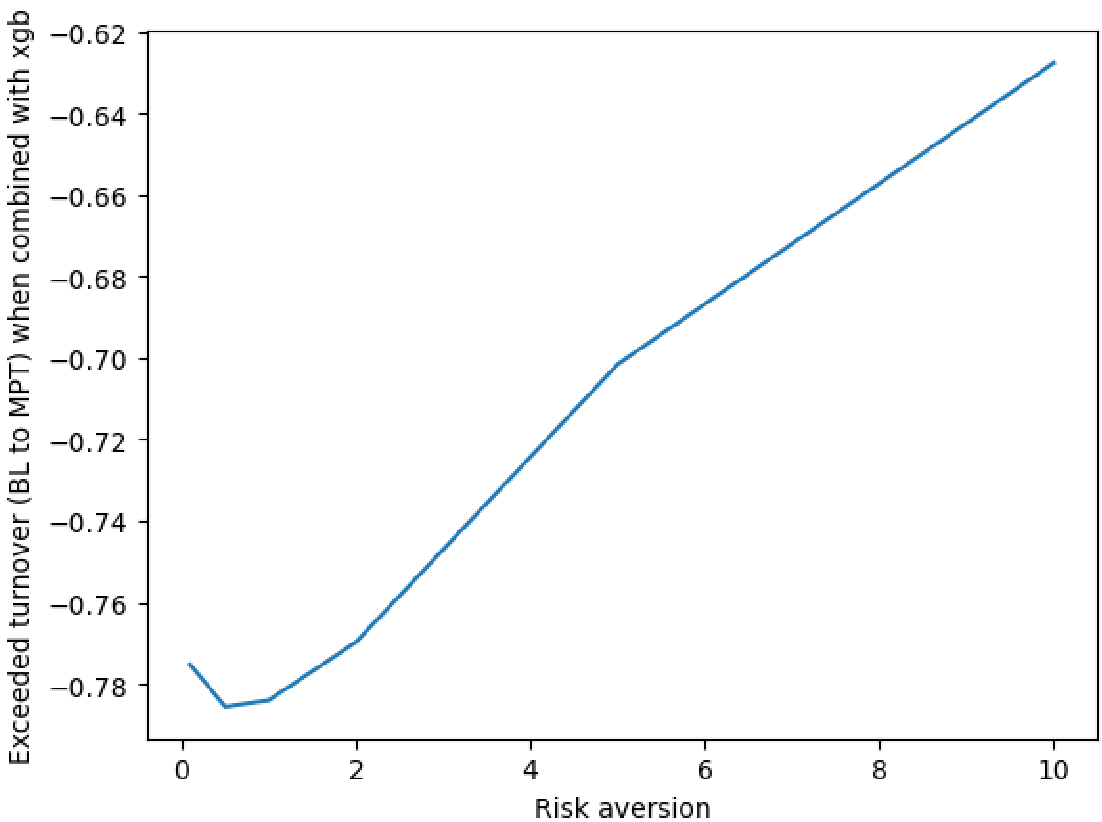
[ ]:
```
plt.figure(dpi = 300)
plt.plot([0.1, 0.5, 1, 2, 5, 10], exceeded_turnover_lgb)
plt.xlabel("Risk aversion")
plt.ylabel("Exceeded turnover (BL to MPT) when combined with lgb")
```

[ ]:
```
Text(0, 0.5, 'Exceeded turnover (BL to MPT) when combined with lgb')
```



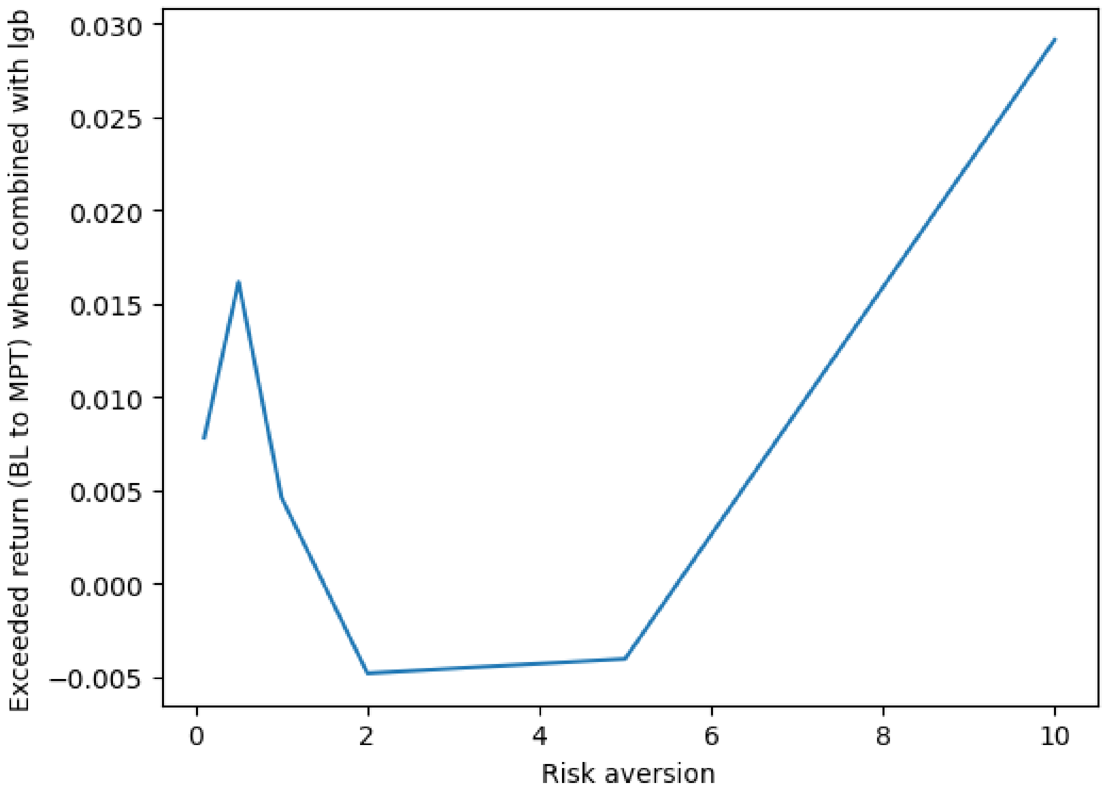
[ ]:
```
plt.plot([0.1, 0.5, 1, 2, 5, 10], exceeded_turnover_xgb)
plt.xlabel("Risk aversion")
plt.ylabel("Exceeded turnover (BL to MPT) when combined with xgb")
```

[ ]: Text(0, 0.5, 'Exceeded turnover (BL to MPT) when combined with xgb')



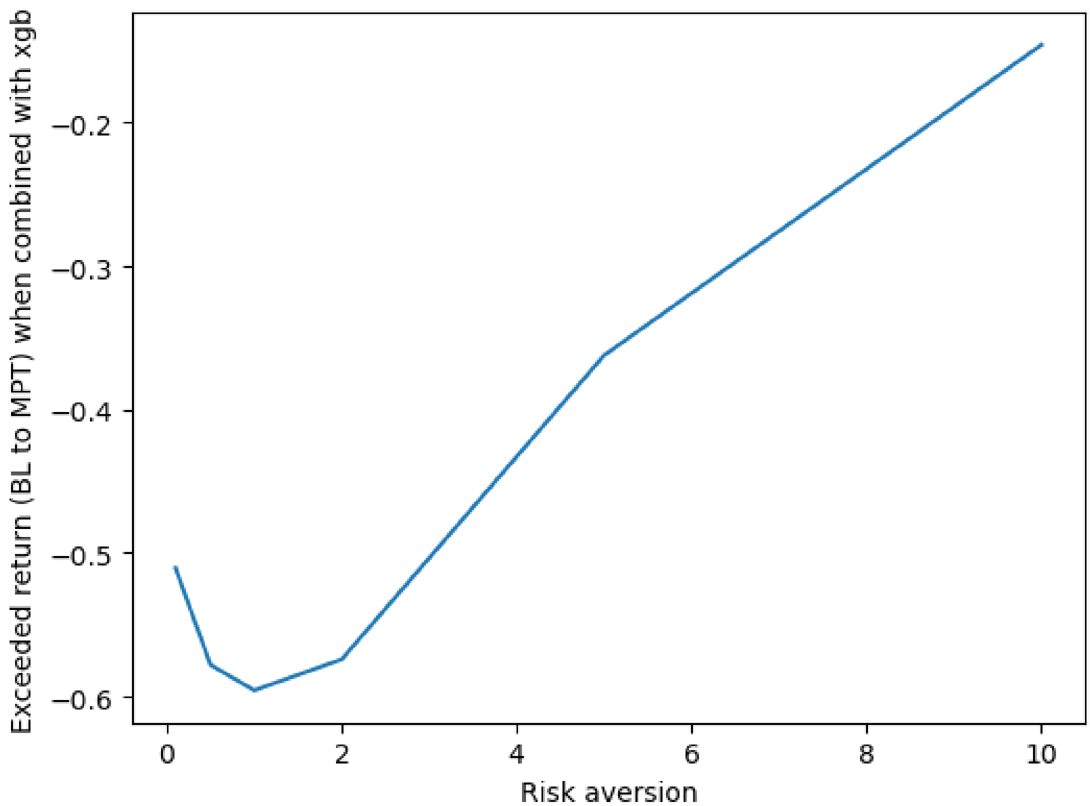
[ ]:
```
plt.plot([0.1, 0.5, 1, 2, 5, 10], exceeded_return_lgb)
plt.xlabel("Risk aversion")
plt.ylabel("Exceeded return (BL to MPT) when combined with lgb")
```

[ ]: Text(0, 0.5, 'Exceeded return (BL to MPT) when combined with lgb')



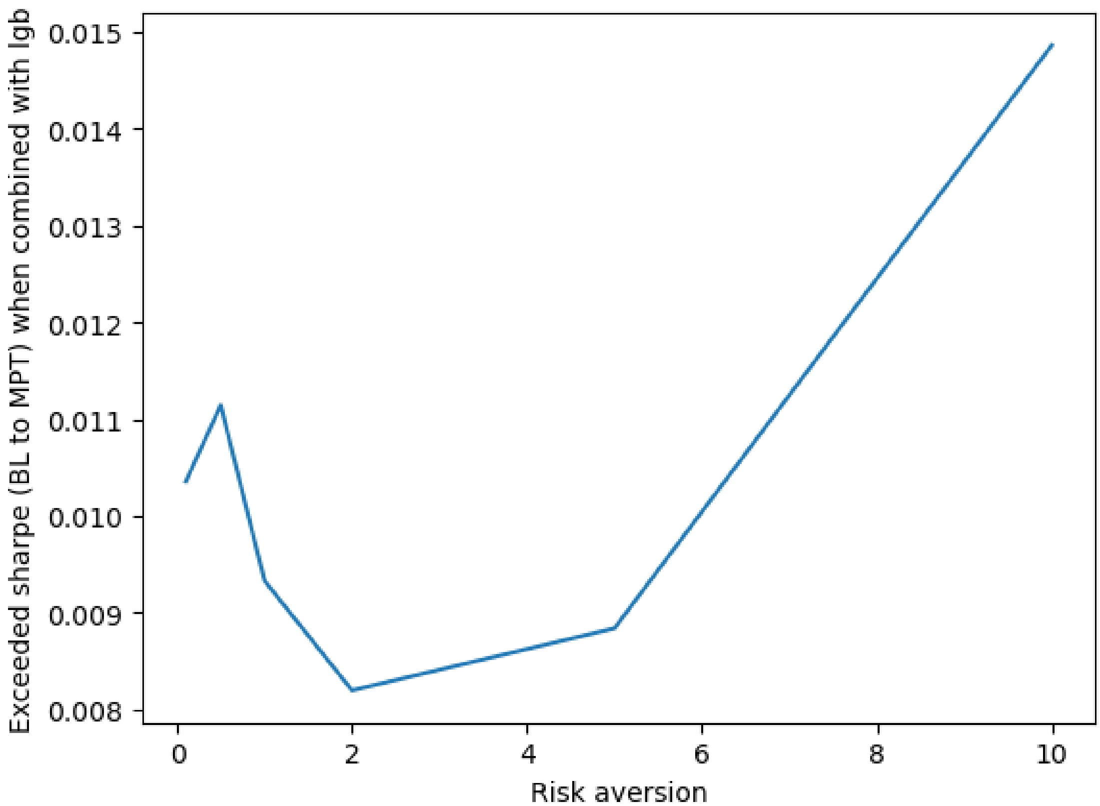
[ ]:
```
plt.plot([0.1, 0.5, 1, 2, 5, 10], exceeded_return_xgb)
plt.xlabel("Risk aversion")
plt.ylabel("Exceeded return (BL to MPT) when combined with xgb")
```

[ ]:
```
Text(0, 0.5, 'Exceeded return (BL to MPT) when combined with xgb')
```



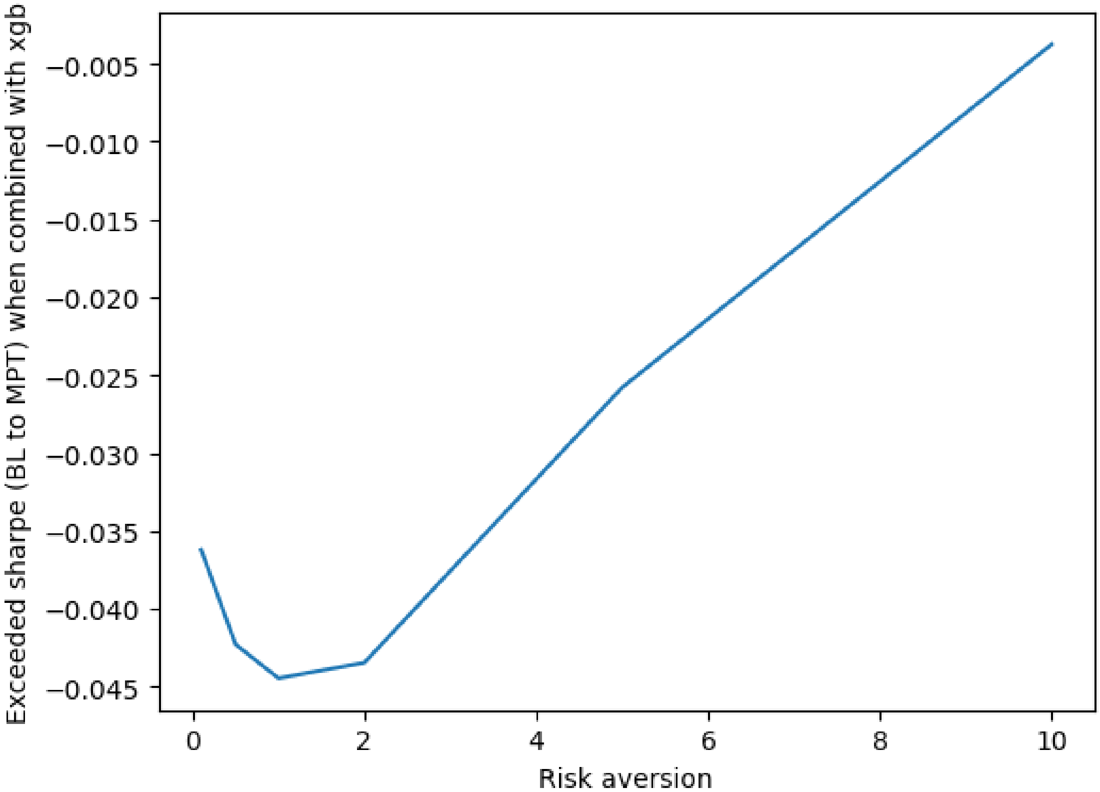
[ ]:
```
plt.plot([0.1, 0.5, 1, 2, 5, 10], exceeded_sharpe_lgb)
plt.xlabel("Risk aversion")
plt.ylabel("Exceeded sharpe (BL to MPT) when combined with lgb")
```

[ ]:
```
Text(0, 0.5, 'Exceeded sharpe (BL to MPT) when combined with lgb')
```



[ ]:
```
plt.plot([0.1, 0.5, 1, 2, 5, 10], exceeded_sharpe_xgb)
plt.xlabel("Risk aversion")
plt.ylabel("Exceeded sharpe (BL to MPT) when combined with xgb")
```

[ ]:
```
Text(0, 0.5, 'Exceeded sharpe (BL to MPT) when combined with xgb')
```


Exceeded sharpe (BL to MPT) when combined with xgb
−0.005
−0.010
−0.015
−0.020
−0.025
−0.030
−0.035
−0.040
−0.045
0
2
4
6
8
10
Risk aversion