A. Python code including parameters used for search:

```
    BayesSearchCV(
            lgb.LGBMClassifier(),
            {
                'n_estimators': [50, 100, 150, 200, 250, 300, 350, 400, 450, 500],
                'max_depth': Integer(1, 70),
                'class_weight': ['balanced', None],

            },
        random_state=42,
        scoring = roc_auc_weighted
   )
```

B. Best parameters for each model externally validated at different time periods for “1-year-survival”

Best Parameters: {'class_weight': 'balanced', 'max_depth': 64, 'n_estimators': 100}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',* *colsample_bytree=1.0, importance_type='split', learning_rate=0.1,* *max_depth=64, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=100, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 428 seconds
Memory Use: peak memory: 793.00 MiB, increment: 581.54 MiB

C. Best parameters for each model externally validated at different time periods for “2-year-survival”

Best Parameters: {'class_weight': 'balanced', 'max_depth': 70, 'n_estimators': 150}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',* *colsample_bytree=1.0, importance_type='split', learning_rate=0.1,* *max_depth=70, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=150, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 568 seconds
Memory Use:

peak memory: 704.86 MiB, increment: 510.96 MiB

D. Best parameters for each model externally validated at different time periods for “3-year-survival”

Best Parameters: {'class_weight': 'balanced', 'max_depth': 64, 'n_estimators': 100}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',* *colsample_bytree=1.0, importance_type='split', learning_rate=0.1,* *max_depth=64, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=100, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 628 seconds
Memory Use: peak memory: 669.03 MiB, increment: 502.62 MiB

E. Best parameters for each model externally validated at different time periods for “4-year-survival”

Best Parameters: {'class_weight': 'balanced', 'max_depth': 64, 'n_estimators': 100}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',* *colsample_bytree=1.0, importance_type='split', learning_rate=0.1,* *max_depth=64, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=100, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 628 seconds
Memory Use: peak memory: 710.25 MiB, increment: 425.25 MiB

F. Best parameters for each model externally validated at different time periods for “5-year-survival”

Best Parameters: {'class_weight': 'balanced', 'max_depth': 70, 'n_estimators': 150}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',*

colsample_bytree=1.0, importance_type='split', learning_rate=0.1, *max_depth=70, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=150, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 480 seconds
Memory Use: peak memory: 600.12 MiB, increment: 369.88 MiB

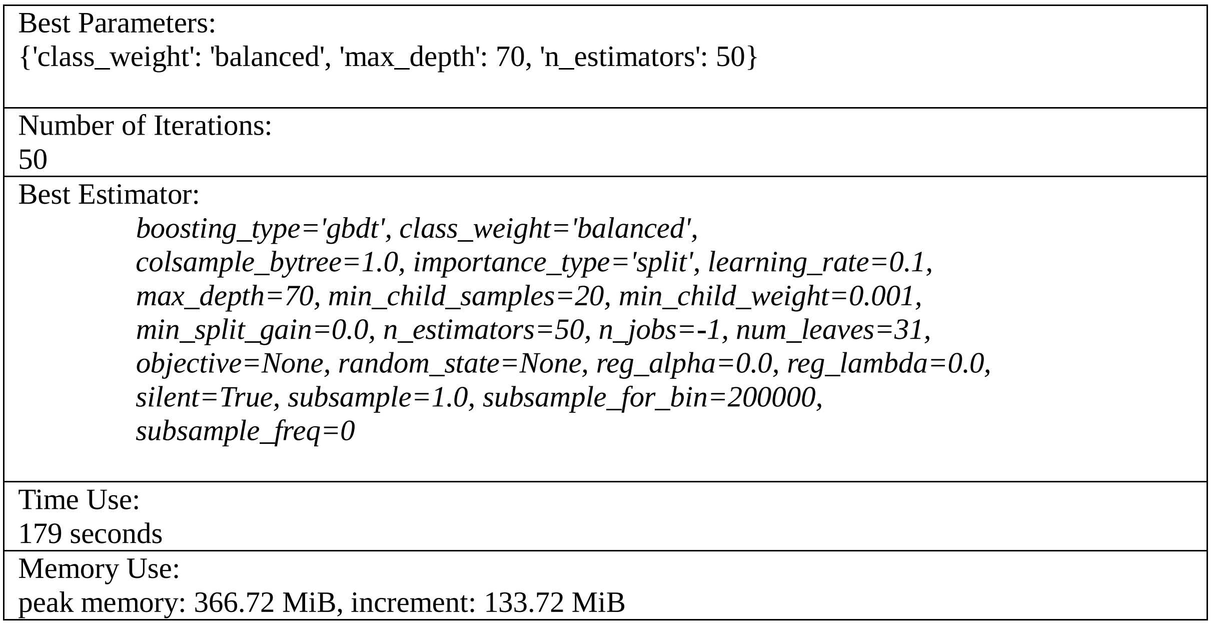
G. Best parameters for each model externally validated at different time periods for “6-year-survival”

Best Parameters: {'class_weight': 'balanced', 'max_depth': 70, 'n_estimators': 150}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',* *colsample_bytree=1.0, importance_type='split', learning_rate=0.1,* *max_depth=70, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=150, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 515 seconds
Memory Use: peak memory: 626.88 MiB, increment: 372.21 MiB

H. Best parameters for each model externally validated at different time periods for “8-year-survival”

Best Parameters: {'class_weight': 'balanced', 'max_depth': 64, 'n_estimators': 100}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',* *colsample_bytree=1.0, importance_type='split', learning_rate=0.1,* *max_depth=64, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=100, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 227 seconds
Memory Use: peak memory: 470.20 MiB, increment: 228.31 MiB

I. Best parameters for each model externally validated at different time periods for “10-year-survival”



Best Parameters: {'class_weight': 'balanced', 'max_depth': 70, 'n_estimators': 50}
Number of Iterations: 50
Best Estimator: *boosting_type='gbdt', class_weight='balanced',* *colsample_bytree=1.0, importance_type='split', learning_rate=0.1,* *max_depth=70, min_child_samples=20, min_child_weight=0.001,* *min_split_gain=0.0, n_estimators=50, n_jobs=-1, num_leaves=31,* *objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,* *silent=True, subsample=1.0, subsample_for_bin=200000,* *subsample_freq=0*
Time Use: 179 seconds
Memory Use: peak memory: 366.72 MiB, increment: 133.72 MiB

S2 Fig. Bayesian optimization for hyperparameter tuning.