

search_model_r2_allplayer

May 27, 2023

```
[ ]: from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.model_selection import cross_val_score
from sklearn import linear_model
import numpy as np
import pandas as pd
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_log_error

[ ]: # player
player = pd.read_csv("../datafrom200/players.csv")
# x_list = player.columns[7:-5]
# x_list = ['Crossing', 'Finishing', 'Heading_Accuracy', 'Short_Passing', 'Volleys',
#           'Dribbling', 'Curve', 'FK_Accuracy', 'Long_Passing', 'Ball_Control',
#           'Acceleration', 'Sprint_Speed', 'Agility', 'Reactions', 'Balance',
#           'Shot_Power', 'Jumping', 'Stamina', 'Strength', 'Long_Shots',
#           'Aggression', 'Interceptions', 'Positioning', 'Vision', 'Penalties',
#           'Composure', 'Defensive_Awareness']
# x_list = ['Ball_Control', 'Sprint_Speed', 'Reactions', 'Stamina', 'Composure',
#           'Standing_Tackle', 'Sliding_Tackle'] # xgboost feature selection
x_list = ['Finishing', 'Short_Passing', 'Dribbling', 'Long_Passing',
          'Ball_Control', 'Acceleration', 'Sprint_Speed', 'Reactions', 'Balance',
          'Shot_Power', 'Stamina', 'Strength', 'Aggression', 'Vision',
          'Penalties', 'Standing_Tackle', 'Sliding_Tackle'] # random forest auto
# feature selection
print(x_list)
X = player[x_list]
y = player["value"]
len(x_list)
```

```
['Finishing', 'Short_Passing', 'Dribbling', 'Long_Passing', 'Ball_Control',
'Acceleration', 'Sprint_Speed', 'Reactions', 'Balance', 'Shot_Power', 'Stamina',
'Strength', 'Aggression', 'Vision', 'Penalties', 'Standing_Tackle',
'Sliding_Tackle']
```

[]: 17

```
[ ]: # train, test : 0.8, 0.2
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↳random_state=21)
# X_validation, X_test, y_validation, y_test = train_test_split(X_test_val,
↳y_test_val, test_size=0.5, random_state=23)
```

```
[ ]: # XGBoost
import xgboost as xgb
model = xgb.XGBRegressor()
params = {'learning_rate': [0.1, 0.2, 0.3],
          'max_depth': np.arange(3, 15, 2),
          # 'min_child_weight' : np.arange(3, 15, 2)
        }

grid_search = GridSearchCV(model, params, cv=10, verbose=2, n_jobs=-1)
grid_search.fit(X_train, y_train)

#
print('Best parameters: ', grid_search.best_params_)
print('Best cross-validation score: ', grid_search.best_score_)

#
best_model = grid_search.best_estimator_
print('Score on test set: ', best_model.score(X_test, y_test))
```

Fitting 10 folds for each of 18 candidates, totalling 180 fits

```
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.6s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.6s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.6s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.7s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.7s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.7s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.7s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.7s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.7s
[CV] END ...learning_rate=0.1, max_depth=3; total time= 0.7s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.3s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.3s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.0s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.0s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.1s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.1s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.1s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.2s
[CV] END ...learning_rate=0.1, max_depth=5; total time= 1.2s
```


[illegible]

```

[CV] END ...learning_rate=0.3, max_depth=11; total time= 3.3s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 3.2s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 4.6s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 3.3s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 4.6s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 3.4s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 3.3s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 3.5s
[CV] END ...learning_rate=0.3, max_depth=11; total time= 4.2s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 4.2s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.8s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 4.4s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.9s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.9s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.7s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.5s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.6s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.5s
[CV] END ...learning_rate=0.3, max_depth=13; total time= 3.6s
Best parameters: {'learning_rate': 0.3, 'max_depth': 3}
Best cross-validation score: 0.8760604744535361
Score on test set: 0.9010435817311566

```

```

[ ]: # Catboost
import catboost
model = catboost.CatBoostRegressor()
params = {'depth' : [16],
          # 'iterations' : [500, 1000, 1500],
          # 'learning_rate' : [0.01, 0.02, 0.03],
          # 'subsample' : [0.7, 0.8, 0.9, 1]
        }

grid_search = GridSearchCV(model, params, cv=5, verbose=2, n_jobs=4)
grid_search.fit(X_train, y_train)

#
print('Best parameters: ', grid_search.best_params_)
print('Best cross-validation score: ', grid_search.best_score_)

#
best_model = grid_search.best_estimator_
print('Score on test set: ', best_model.score(X_test, y_test))

```

```

[ ]: import lightgbm as lgb
model = lgb.LGBMRegressor()
params = {'n_estimators': np.arange(100, 1100, 200),
          'max_depth': np.arange(3, 16, 4),

```

```

        'subsample': [0.7, 0.8, 0.9, 1],
        'colsample_bytree': [0.7, 0.8, 0.9, 1],
        'learning_rate': [1e-5, 1e-3, 1e-1],
        'num_leaves': np.arange(10, 110, 20)}

grid_search = GridSearchCV(model, params, cv=5, verbose=2, n_jobs=-1)
grid_search.fit(X_train, y_train)

#
print('Best parameters: ', grid_search.best_params_)
print('Best cross-validation score: ', grid_search.best_score_)

#
best_model = grid_search.best_estimator_
print('Score on test set: ', best_model.score(X_test, y_test))

```

Fitting 5 folds for each of 4800 candidates, totalling 24000 fits

```

[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.7; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.7; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.7; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.7; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.7; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.7; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.8; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.8; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.8; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.8; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.8; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.9; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.8; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.9; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.9; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.9; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.9; total time= 0.1s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=3,
n_estimators=100, num_leaves=10, subsample=0.9; total time= 0.1s

```



```
n_estimators=700, num_leaves=90, subsample=0.7; total time= 0.5s[CV] END
colsample_bytree=0.7, learning_rate=1e-05, max_depth=3, n_estimators=700,
num_leaves=70, subsample=0.9; total time= 0.9s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
n_estimators=500, num_leaves=30, subsample=0.8; total time= 0.8s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=0.8; total time= 0.9s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=0.9; total time= 0.9s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=0.9; total time= 0.8s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=0.9; total time= 0.8s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=0.9; total time= 0.9s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=1; total time= 0.9s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=1; total time= 0.9s
[CV] END colsample_bytree=0.7, learning_rate=1e-05, max_depth=15,
n_estimators=500, num_leaves=30, subsample=1; total time= 0.9s[CV] END
colsample_bytree=0.7, learning_rate=1e-05, max_depth=15, n_estimators=500,
num_leaves=30, subsample=1; total time= 0.9s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
num_leaves=70, subsample=1; total time= 0.3s
```


[illegible]

```
num_leaves=70, subsample=0.9; total time= 0.4s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
colsample_bytree=0.7, learning_rate=0.001, max_depth=15, n_estimators=100,
num_leaves=70, subsample=0.8; total time= 0.4s
```

```
num_leaves=90, subsample=0.9; total time= 0.4s
```

```
colsample_bytree=0.7, learning_rate=0.001, max_depth=15, n_estimators=300,
num_leaves=10, subsample=0.8; total time= 0.4s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=100,
num_leaves=30, subsample=0.8; total time= 0.1s[CV] END colsample_bytree=0.7,
learning_rate=0.1, max_depth=3, n_estimators=100, num_leaves=30, subsample=0.8;
total time= 0.1s
```



```
total time= 0.1s
```



```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num_leaves=50, subsample=0.8; total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num_leaves=50, subsample=0.9; total time= 0.4s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=50, subsample=0.9; total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=50, subsample=1; total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=50, subsample=1; total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=50, subsample=1; total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.7; total time= 0.4s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.7; total time= 0.4s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.7; total time= 0.6s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.8; total time= 0.4s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num_leaves=70, subsample=0.7: total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.7: total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.8: total time= 0.4s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num_leaves=70, subsample=0.8: total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.8: total time= 0.5s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num_leaves=70, subsample=0.8: total time= 0.4s
```

```
[CV] END colsample_bytree=0.7, learning_rate=0.1, max_depth=3, n_estimators=700,
num leaves=70, subsample=0.9: total time= 0.4s
```

```
[CV] END colsample bytree=0.7, learning rate=0.1, max depth=3, n estimators=700,
```


[illegible]


```
learning_rate=0.1, max_depth=7, n_estimators=100, num_leaves=10, subsample=1;
total time= 0.1s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
colsample_bytree=0.8, learning_rate=1e-05, max_depth=7, n_estimators=300,
num_leaves=90, subsample=0.7; total time= 0.8s
```

```
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.7; total time= 0.5s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
n_estimators=700, num_leaves=30, subsample=0.8; total time= 1.4s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=0.9; total time= 1.3s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=0.8; total time= 1.6s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=0.9; total time= 1.3s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=0.8; total time= 1.6s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=0.9; total time= 1.2s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=1; total time= 1.4s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=1; total time= 1.4s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=0.9; total time= 1.5s
[CV] END colsample_bytree=0.8, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=30, subsample=0.9; total time= 1.7s[CV] END
colsample_bytree=0.8, learning_rate=1e-05, max_depth=11, n_estimators=700,
num_leaves=30, subsample=1; total time= 1.5s
```


[illegible]

[illegible]

[illegible]

[illegible]


```
colsample_bytree=0.8, learning_rate=1e-05, max_depth=15, n_estimators=500,
num_leaves=50, subsample=1; total time= 1.4s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
num_leaves=10, subsample=0.7; total time= 0.6s
```



```
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=3,
n_estimators=500, num_leaves=30, subsample=0.9; total time= 0.5s[CV] END
colsample_bytree=0.8, learning_rate=0.001, max_depth=3, n_estimators=500,
num_leaves=30, subsample=1; total time= 0.4s
```

[illegible]

[illegible]


```
n_estimators=700, num_leaves=50, subsample=0.9; total time= 0.5s[CV] END
colsample_bytree=0.8, learning_rate=0.001, max_depth=3, n_estimators=700,
num_leaves=50, subsample=0.9; total time= 0.5s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
num_leaves=90, subsample=0.7; total time= 1.2s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

colsample_bytree=0.8, learning_rate=0.001, max_depth=15, n_estimators=100,
num_leaves=50, subsample=0.8; total time=    0.4s[CV] END colsample_bytree=0.8,
learning_rate=0.001, max_depth=15, n_estimators=100, num_leaves=50,
subsample=0.9; total time=    0.3s

```

```
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=100, num_leaves=50, subsample=0.9; total time= 0.3s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=100, num_leaves=50, subsample=0.9; total time= 0.3s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=100, num_leaves=50, subsample=0.9; total time= 0.3s[CV] END
colsample_bytree=0.8, learning_rate=0.001, max_depth=15, n_estimators=100,
num_leaves=50, subsample=0.9; total time= 0.4s
```



```
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=30, subsample=1; total time= 0.6s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=30, subsample=1; total time= 0.7s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=30, subsample=1; total time= 0.8s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=50, subsample=0.7; total time= 0.8s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=50, subsample=0.7; total time= 0.9s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=50, subsample=0.7; total time= 1.0s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=50, subsample=0.7; total time= 0.9s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=50, subsample=0.7; total time= 1.0s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=50, subsample=0.8; total time= 0.9s
[CV] END colsample_bytree=0.8, learning_rate=0.001, max_depth=15,
n_estimators=300, num_leaves=50, subsample=0.8; total time= 0.9s[CV] END
colsample_bytree=0.8, learning_rate=0.001, max_depth=15, n_estimators=300,
num_leaves=50, subsample=0.8; total time= 0.8s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

colsample_bytree=0.9, learning_rate=1e-05, max_depth=11, n_estimators=100,
num_leaves=50, subsample=0.8; total time= 0.3s[CV] END colsample_bytree=0.9,
learning_rate=1e-05, max_depth=7, n_estimators=900, num_leaves=90, subsample=1;
total time= 3.3s

[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=0.8; total time= 0.3s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=0.8; total time= 0.3s[CV] END
colsample_bytree=0.9, learning_rate=1e-05, max_depth=11, n_estimators=100,
num_leaves=50, subsample=0.8; total time= 0.4s

[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=0.8; total time= 0.3s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=7,
n_estimators=900, num_leaves=90, subsample=1; total time= 3.3s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=1; total time= 0.3s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=0.9; total time= 0.3s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=1; total time= 0.3s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=1; total time= 0.3s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=50, subsample=1; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=70, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=70, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=70, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=7,
n_estimators=900, num_leaves=90, subsample=1; total time= 3.0s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,
n_estimators=100, num_leaves=70, subsample=0.7; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=1e-05, max_depth=11,

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
n_estimators=900, num_leaves=30, subsample=0.7; total time= 1.9s[CV] END
colsample_bytree=0.9, learning_rate=1e-05, max_depth=15, n_estimators=900,
num_leaves=30, subsample=0.8; total time= 1.7s
```


[illegible]

[illegible]

[illegible]

[illegible]


```
colsample_bytree=0.9, learning_rate=0.001, max_depth=3, n_estimators=300,
num_leaves=70, subsample=0.8; total time= 0.2s
```

[illegible]

[illegible]

[illegible]

[illegible]


```
n_estimators=700, num_leaves=30, subsample=0.8; total time= 0.5s[CV] END
colsample_bytree=0.9, learning_rate=0.001, max_depth=3, n_estimators=700,
num_leaves=30, subsample=0.8; total time= 0.5s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.8; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.8; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=300, num_leaves=90, subsample=1; total time= 1.1s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.9; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.9; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.9; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=0.9; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=1; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.001, max_depth=7,
n_estimators=500, num_leaves=10, subsample=1; total time= 0.5s[CV] END
colsample_bytree=0.9, learning_rate=0.001, max_depth=7, n_estimators=500,
num_leaves=10, subsample=1; total time= 0.5s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
colsample_bytree=0.9, learning_rate=0.001, max_depth=7, n_estimators=900,
num_leaves=10, subsample=0.7; total time= 0.9s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
learning_rate=0.1, max_depth=7, n_estimators=100, num_leaves=50, subsample=0.7;
total time= 0.2s
```

[illegible]

[illegible]

[illegible]

```
total time= 0.3s
```

```
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=0.8; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=1; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=1; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=1; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=0.9; total time= 0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=90, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=1; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=90, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=90, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=300,
num_leaves=70, subsample=1; total time= 0.5s
[CV] END colsample bytree=0.9, learning rate=0.1, max depth=7, n estimators=300,
```

[illegible]

```
num_leaves=10, subsample=0.8; total time=    0.5s
[CV] END colsample_bytree=0.9, learning_rate=0.1, max_depth=7, n_estimators=500,
num_leaves=10, subsample=0.7; total time=    0.6s[CV] END colsample_bytree=0.9,
learning_rate=0.1, max_depth=7, n_estimators=500, num_leaves=10, subsample=0.8;
total time=    0.6s
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
learning_rate=1e-05, max_depth=3, n_estimators=100, num_leaves=70,
subsample=0.9; total time= 0.1s
```

[illegible]


```
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.8; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.8; total time= 0.6s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.7; total time= 0.6s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.8; total time= 0.5s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.9; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=1; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=1; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=0.9; total time= 0.5s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=1; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=10, subsample=1; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=30, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=500,
num_leaves=30, subsample=0.7; total time= 0.4s[CV] END colsample_bytree=1,
learning_rate=1e-05, max_depth=3, n_estimators=500, num_leaves=10, subsample=1;
total time= 0.5s
```


[illegible]

subsample=0.7; total time= 0.7s

```
learning_rate=1e-05, max_depth=3, n_estimators=700, num_leaves=50,
subsample=0.8; total time= 0.5s
```

[illegible]

[illegible]

```

num_leaves=70, subsample=0.8; total time= 1.0s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=0.9; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=0.8; total time= 0.9s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=0.9; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=0.9; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=0.9; total time= 0.9s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=1; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=0.9; total time= 1.1s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=1; total time= 0.7s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=1; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=90, subsample=0.7; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=90, subsample=0.7; total time= 0.6s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=90, subsample=0.7; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=3, n_estimators=900,
num_leaves=70, subsample=1; total time= 1.0s[CV] END colsample_bytree=1,
learning_rate=1e-05, max_depth=3, n_estimators=900, num_leaves=70, subsample=1;
total time= 1.0s

```



```
learning_rate=1e-05, max_depth=7, n_estimators=300, num_leaves=30,
subsample=0.9; total time= 0.6s
```

[illegible]

[illegible]

[illegible]

```
subsample=0.8; total time= 1.0s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

n_estimators=900, num_leaves=10, subsample=0.7; total time= 0.9s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=0.9; total time= 4.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=0.9; total time= 3.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.8; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=0.9; total time= 4.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.8; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.8; total time= 0.9s[CV] END
colsample_bytree=1, learning_rate=1e-05, max_depth=11, n_estimators=900,
num_leaves=10, subsample=0.8; total time= 1.1s

```

```

[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.9; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.8; total time= 1.0s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=1; total time= 4.0s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=1; total time= 4.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.9; total time= 1.0s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.9; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=1; total time= 3.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.9; total time= 1.1s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=1; total time= 4.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=700, num_leaves=90, subsample=1; total time= 5.1s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=1; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=1; total time= 0.8s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=1; total time= 1.0s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=0.9; total time= 1.1s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=11,
n_estimators=900, num_leaves=10, subsample=1; total time= 1.0s

```


[illegible]

[illegible]

```
num_leaves=30, subsample=0.7; total time= 0.3s
```

```
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.8; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.9; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.8; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.9; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.9; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.9; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.9; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=0.9; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=1; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=1; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=1; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=1; total time= 0.2s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=30, subsample=1; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=50, subsample=0.7; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=50, subsample=0.7; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=50, subsample=0.7; total time= 0.4s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
n_estimators=100, num_leaves=50, subsample=0.7; total time= 0.3s
[CV] END colsample_bytree=1, learning_rate=1e-05, max_depth=15,
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
learning_rate=0.001, max_depth=3, n_estimators=300, num_leaves=30, subsample=1;
total time= 0.3s
```

[illegible]

[illegible]

[illegible]


```
[CV] END colsample_bytree=1, learning_rate=0.001, max_depth=7, n_estimators=100,
num_leaves=90, subsample=0.8; total time= 0.3s[CV] END colsample_bytree=1,
learning_rate=0.001, max_depth=7, n_estimators=100, num_leaves=90,
subsample=0.7; total time= 0.5s
```


[illegible]

[illegible]


```
num_leaves=10, subsample=0.7; total time= 0.7s[CV] END colsample_bytree=1,
learning_rate=0.001, max_depth=7, n_estimators=500, num_leaves=90, subsample=1;
total time= 1.7s
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
total time= 0.6s
```


[illegible]

[illegible]

[illegible]


```
num_leaves=50, subsample=0.7; total time= 0.3s[CV] END colsample_bytree=1,
learning_rate=0.1, max_depth=11, n_estimators=900, num_leaves=90, subsample=1;
total time= 2.7s
```

[illegible]

[illegible]


```
[CV] END colsample_bytree=1, learning_rate=0.1, max_depth=15, n_estimators=900,
num_leaves=90, subsample=1; total time= 2.1s
[CV] END colsample_bytree=1, learning_rate=0.1, max_depth=15, n_estimators=900,
num_leaves=90, subsample=1; total time= 2.1s
[CV] END colsample_bytree=1, learning_rate=0.1, max_depth=15, n_estimators=900,
num_leaves=90, subsample=1; total time= 2.0s
[CV] END colsample_bytree=1, learning_rate=0.1, max_depth=15, n_estimators=900,
num_leaves=90, subsample=1; total time= 1.8s
Best parameters: {'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth':
7, 'n_estimators': 900, 'num_leaves': 30, 'subsample': 0.7}
Best cross-validation score: 0.8784135655430839
Score on test set: 0.8894490540930118
```

```
[ ]: #
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
params = {'criterion': ['gini', 'entropy'],
          'max_depth': np.arange(3, 16, 2),
          'max_features': ['auto', 'sqrt', 'log2'],
          'min_samples_leaf': np.arange(3, 15, 2),
          'min_samples_split': np.arange(3, 15, 2),
          'n_estimators': np.arange(100, 1100, 100) }

grid_search = GridSearchCV(model, params, cv=10, verbose=3, n_jobs=-1)
grid_search.fit(X_train, y_train)

#
print('Best parameters: ', grid_search.best_params_)
print('Best cross-validation score: ', grid_search.best_score_)

#
best_model = grid_search.best_estimator_
print('Score on test set: ', best_model.score(X_test, y_test))
```

-----random forest starts----- [Parallel(n_jobs=-1)]: Done 7560 out of 7560 |
elapsed: 3.0min finished ('Best parameters:', {'max_features': 'auto', 'min_samples_split': 5,
'max_depth': 13, 'min_samples_leaf': 3}) ('Best cross-validation score:', 0.8485611469873839)
('Score on test set:', 0.8261229514037156) -----random forest ends-----

```
[ ]: # GBDT
from sklearn.ensemble import GradientBoostingRegressor
model = GradientBoostingRegressor()
params = {'n_estimators': [50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000],
          'max_depth': [1, 2, 3, 5, 7, 10],
          # 'min_samples_split': np.arange(3, 16, 2),
          # 'min_samples_leaf': np.arange(3, 16, 2),
```

```

        # 'subsample': np.arange(0.7, 1, 0.05),
        'learning_rate': [1e-2, 1e-1]
    }
grid_search = GridSearchCV(model, params, cv=5, verbose=1, n_jobs=-1)
grid_search.fit(X_train, y_train)

#
print('Best parameters: ', grid_search.best_params_)
print('Best cross-validation score: ', grid_search.best_score_)

#
best_model = grid_search.best_estimator_
print('Score on test set: ', best_model.score(X_test, y_test))

```

—————-GBDT starts—————- Fitting 10 folds for each of 132 candidates, totalling 1320 fits [Parallel(n_jobs=-1)]: Done 132 tasks | elapsed: 5.1s [Parallel(n_jobs=-1)]: Done 437 tasks | elapsed: 44.5s [Parallel(n_jobs=-1)]: Done 787 tasks | elapsed: 4.7min [Parallel(n_jobs=-1)]: Done 1320 out of 1320 | elapsed: 11.2min finished ('Best parameters:', {'n_estimators': 800, 'learning_rate': 0.1, 'max_depth': 5}) ('Best cross-validation score:', 0.8841836610071835) ('Score on test set:', 0.8927505622156233) —————-GBDT ends—————-

```

[ ]: # Adaboost
from sklearn.ensemble import AdaBoostRegressor
model = AdaBoostRegressor()
params = {
    'n_estimators': np.arange(50, 1100, 100),
    'learning_rate': [0.1, 0.5, 1],
    'loss': ['linear', 'square', 'exponential']
}
grid_search = GridSearchCV(model, params, cv=10, verbose=1, n_jobs=-1)
grid_search.fit(X_train, y_train)

#
print('Best parameters: ', grid_search.best_params_)
print('Best cross-validation score: ', grid_search.best_score_)

#
best_model = grid_search.best_estimator_
print('Score on test set: ', best_model.score(X_test, y_test))

```

—————-Adaboost starts—————- Fitting 10 folds for each of 99 candidates, totalling 990 fits [Parallel(n_jobs=-1)]: Done 104 tasks | elapsed: 25.8s [Parallel(n_jobs=-1)]: Done 354 tasks | elapsed: 1.5min [Parallel(n_jobs=-1)]: Done 704 tasks | elapsed: 2.7min [Parallel(n_jobs=-1)]: Done 990 out of 990 | elapsed: 3.8min finished ('Best parameters:', {'n_estimators': 150, 'loss': 'square', 'learning_rate': 0.1}) ('Best cross-validation score:', 0.7726255786280866) ('Score on test set:', 0.7546916447363856) —————-Adaboost ends—————-


```
[ ]: # # Adaboost
# from sklearn.ensemble import AdaBoostRegressor
# i = 10
# clf = AdaBoostRegressor(n_estimators=i)
# clf.fit(X_train, y_train)
# print("R", clf.score(X_validation, y_validation))
# print("RMSE", np.sqrt(mean_squared_error(clf.predict(X_validation),
# ↪y_validation)))
# print("MAE", mean_absolute_error(clf.predict(X_validation), y_validation))
# print("RMSLE", mean_squared_log_error(clf.predict(X_validation),
# ↪y_validation))
```

R 0.7729781975502361
 RMSE 4286431.008031562
 MAE 2043267.1213696746
 RMSLE 1.5085553838770587

```
[ ]: # list_adaboost # adaboost: 10, 0.6945326685800794
# n_list_lbm = np.array(list_lightgbm) # leaves, depth, n, score : 31, 19, 500,
# ↪0.8007898902056965
# list_randomforest # [80, 0.7839175337533079]
# list_gbdt # [n = 600, 0.8264955831199144]
# catboost default: 0.8027687798692258
# list_xgboost # [60, 20, 0.7812782293507701]
```

1 Adaboost

0.7678737783884331 R
 4334351.772382652 RMSE
 2034372.1610311233 MAE
 1.4164760588279637 RMSLE

2 Xgboost

0.8868637133128766 R
 3025958.826846494 RMSE
 923966.9140471614 MAE
 0.15334408868795865 RMSLE

3 lgb

0.8948018678333571 R
 2917870.8720710855 RMSE

863827.241753701 MAE

4

0.8810732244650263 R

3102429.1779398853 RMSE

971012.3118455497 MAE

0.15934332311326996 RMSLE

5 GBDT

0.900796741780507 R

2833511.7791834464 RMSE

948986.9158787137 MAE

6 Catboost

0.8959925992883182 R

2901310.272421131 RMSE

879135.1904306137 MAE