Hybrid_Model

February 25, 2025

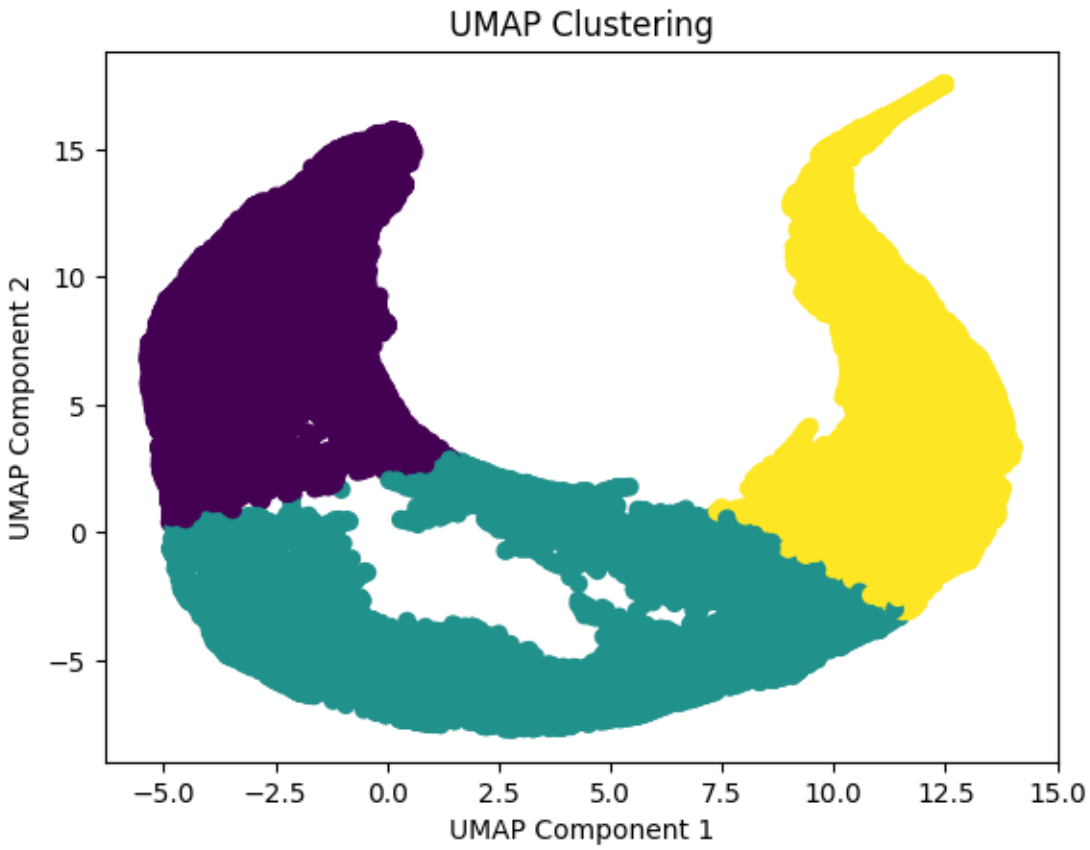
Preparing Data And Preprocessing

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.decomposition import PCA
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler, PowerTransformer
from sklearn.feature_selection import VarianceThreshold
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.neighbors import LocalOutlierFactor
from kneed import KneeLocator
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
import umap.umap_ as umap

# ---- STEP 1: LOAD DATA ----
df = pd.read_excel('./data_files/Data_re.xlsx')  # Ensure the file exists

# ---- STEP 2: FEATURE SELECTION ----
cols_to_keep = ['g_flux', 'r_flux', 'i_flux', 'y_flux', 'z_flux',␣
 ↪'specz_redshift']
df_selected = df[cols_to_keep].copy()

# Compute Flux Color Indices
df_selected['g_r'] = df_selected['g_flux'] - df_selected['r_flux']
df_selected['r_i'] = df_selected['r_flux'] - df_selected['i_flux']
df_selected['i_y'] = df_selected['i_flux'] - df_selected['y_flux']
df_selected['y_z'] = df_selected['y_flux'] - df_selected['z_flux']

# ---- STEP 3: REMOVE OUTLIERS ----
lof = LocalOutlierFactor(n_neighbors=20, contamination=0.02)
outlier_scores = lof.fit_predict(df_selected.drop(columns=['specz_redshift']))
df_clean = df_selected[outlier_scores == 1].copy()
```

```
# ---- STEP 4: SCALING & TRANSFORMING ----
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_clean.drop(columns=['specz_redshift']))

# Apply Power Transformation (Yeo-Johnson for normalizing skewed data)
power_transformer = PowerTransformer(method='yeo-johnson')
df_transformed = power_transformer.fit_transform(df_scaled)

# ---- STEP 5: DIMENSIONALITY REDUCTION WITH UMAP ----
reducer = umap.UMAP(n_components=2, n_neighbors=15, min_dist=0.1,␣
 ↪random_state=42)
X_umap = reducer.fit_transform(df_transformed)

# ---- STEP 6: OPTIONAL PCA FOR EXPLORATION ----
pca = PCA(n_components=3)
X_pca = pca.fit_transform(df_transformed)

# ---- STEP 7: CLUSTERING ----
kmeans = KMeans(n_clusters=3, random_state=42)
df_clean['Cluster'] = kmeans.fit_predict(X_umap)

# ---- PLOT RESULTS ----
plt.scatter(X_umap[:, 0], X_umap[:, 1], c=df_clean['Cluster'], cmap='viridis')
plt.xlabel('UMAP Component 1')
plt.ylabel('UMAP Component 2')
plt.title('UMAP Clustering')
plt.show()
```

```
2025-02-25 15:37:04.291343: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
WARNING: All log messages before absl::InitializeLog() is called are written to
STDERR
E0000 00:00:1740478024.342857    6224 cuda_dnn.cc:8310] Unable to register cuDNN
factory: Attempting to register factory for plugin cuDNN when one has already
been registered
E0000 00:00:1740478024.357959    6224 cuda_blas.cc:1418] Unable to register
cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has
already been registered
2025-02-25 15:37:04.483880: I tensorflow/core/platform/cpu_feature_guard.cc:210]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild
TensorFlow with the appropriate compiler flags.
/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/deprecation.py:151: FutureWarning: 'force_all_finite' was
renamed to 'ensure_all_finite' in 1.6 and will be removed in 1.8.
```

```
  warnings.warn(
/home/chloy/miniconda3/lib/python3.10/site-packages/umap/umap_.py:1952:
UserWarning: n_jobs value 1 overridden to 1 by setting random_state. Use no seed
for parallelism.
  warn(
```



checking

[2]:
```
print(df.head())
print(df.info())
print(df.describe())
```

```
                                                coord       dec  \
0   b'(179325.3125, 99694.8046875, -21178.96484375)' -5.893433
1  b'(179236.609375, 99349.8203125, -23431.181640… -6.522742
2     b'(179281.5, 99283.5078125, -23368.759765625)' -6.505290
3  b'(179365.171875, 99158.046875, -23259.0839843… -6.474628
4      b'(179366.421875, 99172.25, -23188.84765625)' -6.454993

   g_central_image_pop_10px_rad  g_central_image_pop_15px_rad  \
0                             1                             1
```

```
1                              1                              1
2                              1                              1
3                              1                              1
4                              1                              1

   g_central_image_pop_5px_rad  g_cmodel_mag  g_cmodel_magsigma  \
0                            1     20.314907           0.002624
1                            1     22.217360           0.010902
2                            1     21.148739           0.008013
3                            1     18.464205           0.001740
4                            1     20.998287           0.006011

   g_ellipticity  g_half_light_radius  g_isophotal_area  …  z_minor_axis  \
0          0.147                6.047               603  …         4.938
1          0.130                3.430                93  …         2.713
2          0.209                6.597               254  …         4.351
3          0.525               10.855              1064  …         4.815
4          0.738                8.261               386  …         2.279

   z_peak_surface_brightness  z_petro_rad  z_pos_angle  z_sersic_index  \
0                    -8.2933         5.28        36.15           2.193
1                    -7.3657         5.94       -61.78           1.649
2                    -7.6539         9.24        32.76           2.364
3                    -8.5825         6.60        53.15           1.494
4                    -6.8798         5.94        21.16           1.063

         g_flux        r_flux        i_flux        y_flux        z_flux
0  7.482335e-09  2.987891e-08  5.117533e-08  8.117985e-08  6.737572e-08
1  1.297347e-09  5.643225e-09  1.319162e-08  2.141019e-08  1.828079e-08
2  3.471398e-09  1.571304e-08  3.743521e-08  6.504857e-08  5.585498e-08
3  4.114510e-08  1.090914e-07  1.841788e-07  3.080686e-07  2.402212e-07
4  3.987357e-09  1.373544e-08  2.634882e-08  4.449762e-08  3.527128e-08

[5 rows x 89 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40914 entries, 0 to 40913
Data columns (total 89 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   coord                          40914 non-null  object
 1   dec                            40914 non-null  float64
 2   g_central_image_pop_10px_rad   40914 non-null  int64
 3   g_central_image_pop_15px_rad   40914 non-null  int64
 4   g_central_image_pop_5px_rad    40914 non-null  int64
 5   g_cmodel_mag                   40914 non-null  float64
 6   g_cmodel_magsigma              40914 non-null  float64
 7   g_ellipticity                  40914 non-null  float64
 8   g_half_light_radius            40914 non-null  float64
```

```
9   g_isophotal_area              40914 non-null  int64
10  g_major_axis                  40914 non-null  float64
11  g_minor_axis                  40914 non-null  float64
12  g_peak_surface_brightness     40914 non-null  float64
13  g_petro_rad                   40914 non-null  float64
14  g_pos_angle                   40914 non-null  float64
15  g_sersic_index                40914 non-null  float64
16  i_central_image_pop_10px_rad  40914 non-null  int64
17  i_central_image_pop_15px_rad  40914 non-null  int64
18  i_central_image_pop_5px_rad   40914 non-null  int64
19  i_cmodel_mag                  40914 non-null  float64
20  i_cmodel_magsigma             40914 non-null  float64
21  i_ellipticity                 40914 non-null  float64
22  i_half_light_radius           40914 non-null  float64
23  i_isophotal_area              40914 non-null  int64
24  i_major_axis                  40914 non-null  float64
25  i_minor_axis                  40914 non-null  float64
26  i_peak_surface_brightness     40914 non-null  float64
27  i_petro_rad                   40914 non-null  float64
28  i_pos_angle                   40914 non-null  float64
29  i_sersic_index                40914 non-null  float64
30  object_id                     40914 non-null  int64
31  r_central_image_pop_10px_rad  40914 non-null  int64
32  r_central_image_pop_15px_rad  40914 non-null  int64
33  r_central_image_pop_5px_rad   40914 non-null  int64
34  r_cmodel_mag                  40914 non-null  float64
35  r_cmodel_magsigma             40914 non-null  float64
36  r_ellipticity                 40914 non-null  float64
37  r_half_light_radius           40914 non-null  float64
38  r_isophotal_area              40914 non-null  int64
39  r_major_axis                  40914 non-null  float64
40  r_minor_axis                  40914 non-null  float64
41  r_peak_surface_brightness     40914 non-null  float64
42  r_petro_rad                   40914 non-null  float64
43  r_pos_angle                   40914 non-null  float64
44  r_sersic_index                40914 non-null  float64
45  ra                            40914 non-null  float64
46  skymap_id                     40914 non-null  int64
47  specz_dec                     40914 non-null  float64
48  specz_flag_homogeneous        40914 non-null  bool
49  specz_mag_i                   40914 non-null  float64
50  specz_name                    40914 non-null  object
51  specz_ra                      40914 non-null  float64
52  specz_redshift                40914 non-null  float64
53  specz_redshift_err            40914 non-null  float64
54  x_coord                       40914 non-null  float64
55  y_central_image_pop_10px_rad  40914 non-null  int64
56  y_central_image_pop_15px_rad  40914 non-null  int64
```

```
 57  y_central_image_pop_5px_rad   40914 non-null  int64
 58  y_cmodel_mag                  40914 non-null  float64
 59  y_cmodel_magsigma             40914 non-null  float64
 60  y_coord                       40914 non-null  float64
 61  y_ellipticity                 40914 non-null  float64
 62  y_half_light_radius           40914 non-null  float64
 63  y_isophotal_area              40914 non-null  int64
 64  y_major_axis                  40914 non-null  float64
 65  y_minor_axis                  40914 non-null  float64
 66  y_peak_surface_brightness     40914 non-null  float64
 67  y_petro_rad                   40914 non-null  float64
 68  y_pos_angle                   40914 non-null  float64
 69  y_sersic_index                40914 non-null  float64
 70  z_central_image_pop_10px_rad  40914 non-null  int64
 71  z_central_image_pop_15px_rad  40914 non-null  int64
 72  z_central_image_pop_5px_rad   40914 non-null  int64
 73  z_cmodel_mag                  40914 non-null  float64
 74  z_cmodel_magsigma             40914 non-null  float64
 75  z_ellipticity                 40914 non-null  float64
 76  z_half_light_radius           40914 non-null  float64
 77  z_isophotal_area              40914 non-null  int64
 78  z_major_axis                  40914 non-null  float64
 79  z_minor_axis                  40914 non-null  float64
 80  z_peak_surface_brightness     40914 non-null  float64
 81  z_petro_rad                   40914 non-null  float64
 82  z_pos_angle                   40914 non-null  float64
 83  z_sersic_index                40914 non-null  float64
 84  g_flux                        40914 non-null  float64
 85  r_flux                        40914 non-null  float64
 86  i_flux                        40914 non-null  float64
 87  y_flux                        40914 non-null  float64
 88  z_flux                        40914 non-null  float64
dtypes: bool(1), float64(64), int64(22), object(2)
memory usage: 27.5+ MB
None
                dec  g_central_image_pop_10px_rad  \
count  40914.000000                   40914.000000
mean       4.772650                       1.016278
std       14.789896                       0.202830
min       -7.217183                       0.000000
25%       -0.960840                       1.000000
50%        0.311508                       1.000000
75%        1.712535                       1.000000
max       53.260936                       3.000000

       g_central_image_pop_15px_rad  g_central_image_pop_5px_rad  \
count                  40914.000000                 40914.000000
mean                       1.032141                     0.998142
```

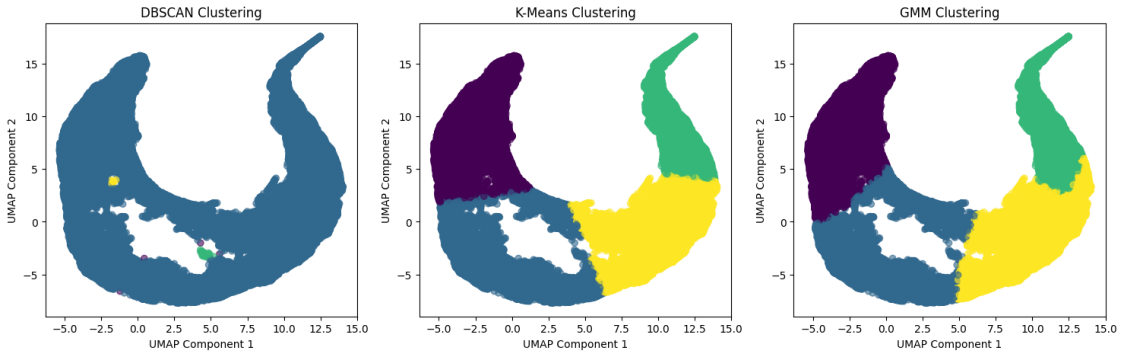
```
std                      0.240109                  0.160037
min                      0.000000                  0.000000
25%                      1.000000                  1.000000
50%                      1.000000                  1.000000
75%                      1.000000                  1.000000
max                      4.000000                  3.000000

       g_cmodel_mag  g_cmodel_magsigma  g_ellipticity  g_half_light_radius  \
count  40914.000000       40914.000000   40914.000000         40914.000000
mean      21.260192           0.010414       0.230579             6.509069
std        1.882030           0.091981       0.171495             3.625851
min       14.753462           0.000149       0.000000             0.000000
25%       19.733286           0.001531       0.095000             3.905000
50%       21.492703           0.004890       0.190000             5.679000
75%       22.614446           0.011530       0.328000             7.976000
max       30.276308          17.593214       0.896000            26.335000

       g_isophotal_area  g_major_axis  …  z_minor_axis  \
count      40914.000000  40914.000000  …  40914.000000
mean         694.781786      5.523996  …      4.064275
std          905.508200      4.095328  …      2.145741
min            0.000000      0.000000  …      0.000000
25%          122.000000      2.706250  …      2.308250
50%          262.000000      3.938000  …      3.740000
75%          947.000000      7.249000  …      5.250000
max         6434.000000     33.320000  …     16.530000

       z_peak_surface_brightness   z_petro_rad   z_pos_angle  z_sersic_index  \
count                40914.000000  40914.000000  40914.000000    40914.000000
mean                    -6.819224      6.249483      1.738910        1.664449
std                      1.418684      1.392911     52.237031        0.870328
min                    -10.477100      0.000000    -89.990000        0.000000
25%                     -7.830100      5.280000    -43.900000        1.037000
50%                     -7.000600      5.940000      3.525000        1.523000
75%                     -5.935925      6.600000     46.870000        2.095000
max                      0.000000     10.560000     90.000000        9.974000

             g_flux        r_flux        i_flux        y_flux        z_flux
count  4.091400e+04  4.091400e+04  4.091400e+04  4.091400e+04  4.091400e+04
mean   1.210340e-08  2.768307e-08  4.473075e-08  6.989669e-08  5.870488e-08
std    2.736938e-08  5.694149e-08  8.773860e-08  1.366205e-07  1.134256e-07
min    7.753125e-13  1.954729e-12  8.507724e-12  3.916208e-11  3.854886e-11
25%    8.999569e-10  2.390220e-09  4.067133e-09  6.001859e-09  5.229988e-09
50%    2.528826e-09  7.571469e-09  1.592592e-08  2.686640e-08  2.237941e-08
75%    1.278456e-08  3.078038e-08  4.926195e-08  7.639837e-08  6.444087e-08
max    1.254918e-06  2.506786e-06  3.731277e-06  5.268088e-06  4.333962e-06

[8 rows x 86 columns]
```

Clustering (DBScan, K means and Gaussian Mixture Method)

[3]:
```
from sklearn.cluster import DBSCAN, KMeans
from sklearn.mixture import GaussianMixture
from sklearn.neighbors import NearestNeighbors
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
import numpy as np
from kneed import KneeLocator

# ---- STEP 6: DETERMINE OPTIMAL DBSCAN EPS ----
k = 5  # Typically, min_samples value
nearest_neighbors = NearestNeighbors(n_neighbors=k)
nearest_neighbors.fit(X_umap)
distances, indices = nearest_neighbors.kneighbors(X_umap)

# Sort distances to find the "knee" point
distances = np.sort(distances[:, -1])

# Use KneeLocator to find optimal epsilon
knee_locator = KneeLocator(range(len(distances)), distances, curve="convex",␣
 ↪direction="increasing")
optimal_eps = distances[knee_locator.elbow]
print(f"Optimal eps for DBSCAN: {optimal_eps:.3f}")

# ---- STEP 6A: APPLY DBSCAN CLUSTERING WITH OPTIMAL EPS ----
dbscan = DBSCAN(eps=optimal_eps, min_samples=k, metric='euclidean')
cluster_labels_dbscan = dbscan.fit_predict(X_umap)
df_clean.loc[:, 'cluster_dbscan'] = cluster_labels_dbscan

# ---- STEP 6B: APPLY K-MEANS & GMM ----
inertia = []
silhouette_scores = []
k_range = range(2, 10)

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init='auto')
    labels = kmeans.fit_predict(X_umap)
    inertia.append(kmeans.inertia_)
    score = silhouette_score(X_umap, labels)
    silhouette_scores.append(score)

knee_locator = KneeLocator(k_range, inertia, curve="convex",␣
 ↪direction="decreasing")
optimal_k = knee_locator.elbow
print(f"Optimal k for K-Means/GMM: {optimal_k}")
```

```
# Apply K-Means with the optimal k
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init='auto')
df_clean.loc[:, 'cluster_kmeans'] = kmeans.fit_predict(X_umap)

# Apply Gaussian Mixture Model (GMM)
gmm = GaussianMixture(n_components=optimal_k, random_state=42)
df_clean.loc[:, 'cluster_gmm'] = gmm.fit_predict(X_umap)

# ---- STEP 7: COMPUTE SILHOUETTE SCORES ----
silhouette_kmeans = silhouette_score(X_umap, df_clean['cluster_kmeans'])
silhouette_gmm = silhouette_score(X_umap, df_clean['cluster_gmm'])
silhouette_dbscan = silhouette_score(X_umap[df_clean['cluster_dbscan'] != -1],
                                     ␣
 ↪df_clean['cluster_dbscan'][df_clean['cluster_dbscan'] != -1])

print(f"Silhouette Scores - KMeans: {silhouette_kmeans:.3f}, GMM:␣
 ↪{silhouette_gmm:.3f}, DBSCAN: {silhouette_dbscan:.3f}")

# ---- STEP 8: VISUALIZE CLUSTERING METHODS (2D PLOTS) ----
fig, ax = plt.subplots(1, 3, figsize=(18, 5))

# DBSCAN 2D Plot
ax[0].scatter(X_umap[:, 0], X_umap[:, 1], c=cluster_labels_dbscan,␣
 ↪cmap='viridis', alpha=0.6)
ax[0].set_title("DBSCAN Clustering")
ax[0].set_xlabel("UMAP Component 1")
ax[0].set_ylabel("UMAP Component 2")

# K-Means 2D Plot
ax[1].scatter(X_umap[:, 0], X_umap[:, 1], c=df_clean['cluster_kmeans'],␣
 ↪cmap='viridis', alpha=0.6)
ax[1].set_title("K-Means Clustering")
ax[1].set_xlabel("UMAP Component 1")
ax[1].set_ylabel("UMAP Component 2")

# GMM 2D Plot
ax[2].scatter(X_umap[:, 0], X_umap[:, 1], c=df_clean['cluster_gmm'],␣
 ↪cmap='viridis', alpha=0.6)
ax[2].set_title("GMM Clustering")
ax[2].set_xlabel("UMAP Component 1")
ax[2].set_ylabel("UMAP Component 2")

plt.show()
```

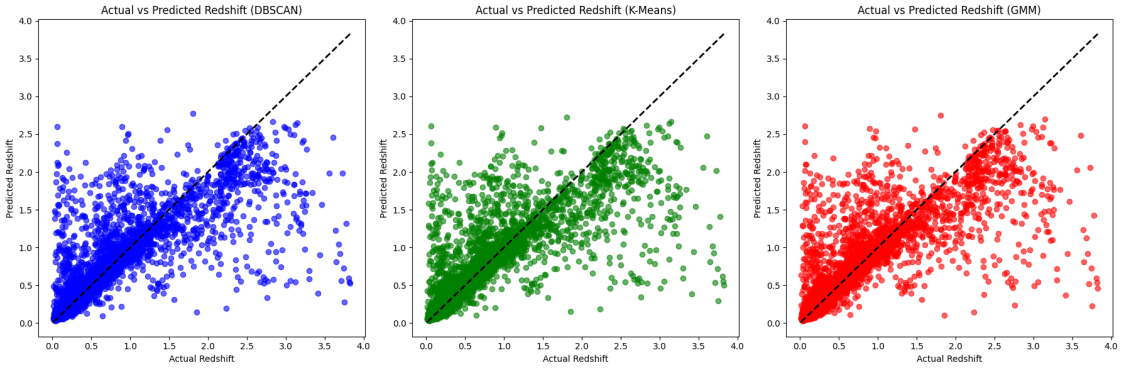
```
Optimal eps for DBSCAN: 0.323
Optimal k for K-Means/GMM: 4
Silhouette Scores - KMeans: 0.486, GMM: 0.455, DBSCAN: -0.422
```



Random Forest

[4]:
```
# ---- STEP 8: RANDOM FOREST REGRESSION ----
# Train separate Random Forest models for each clustering method using flux␣
 ↪features

results = {}

for cluster_type in ['cluster_dbscan', 'cluster_kmeans', 'cluster_gmm']:
    df_temp = pd.get_dummies(df_clean, columns=[cluster_type],␣
 ↪prefix=[f'clust_{cluster_type}'])
    X = df_temp[['g_flux', 'r_flux', 'i_flux', 'y_flux', 'z_flux'] +
                [col for col in df_temp.columns if col.
 ↪startswith(f'clust_{cluster_type}')]]
    y = df_temp['specz_redshift']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

    rf = RandomForestRegressor(n_estimators=100, random_state=42)
    rf.fit(X_train, y_train)

    y_pred = rf.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    results[cluster_type] = {'MSE': mse, 'R2': r2, 'y_pred': y_pred}

# ---- STEP 9: VISUALIZE REGRESSION RESULTS ----
fig, ax = plt.subplots(1, 3, figsize=(18, 6))
titles = ['DBSCAN', 'K-Means', 'GMM']
colors = ['blue', 'green', 'red']
```

```
for i, cluster_type in enumerate(results.keys()):
    ax[i].scatter(y_test, results[cluster_type]['y_pred'], alpha=0.6,␣
 ↪color=colors[i])
    ax[i].set_xlabel('Actual Redshift')
    ax[i].set_ylabel('Predicted Redshift')
    ax[i].set_title(f'Actual vs Predicted Redshift ({titles[i]})')
    ax[i].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],␣
 ↪'k--', lw=2)

plt.tight_layout()
plt.show()

# ---- STEP 10: COMPARE RESULTS ACROSS CLUSTERING METHODS ----
print("Random Forest Regression Results:")
for cluster_type, metrics in results.items():
    print(f"{cluster_type.upper()} - MSE: {metrics['MSE']:.6f}, R2:␣
 ↪{metrics['R2']:.6f}")
```




```
Random Forest Regression Results:
CLUSTER_DBSCAN - MSE: 0.103494, R2: 0.673538
CLUSTER_KMEANS - MSE: 0.103177, R2: 0.674537
CLUSTER_GMM - MSE: 0.103596, R2: 0.673214
```

Checking

[5]:
```
print(df_clean.head())
print(df_clean.columns)
```

```
         g_flux        r_flux        i_flux        y_flux        z_flux  \
0  7.482335e-09  2.987891e-08  5.117533e-08  8.117985e-08  6.737572e-08
1  1.297347e-09  5.643225e-09  1.319162e-08  2.141019e-08  1.828079e-08
2  3.471398e-09  1.571304e-08  3.743521e-08  6.504857e-08  5.585498e-08
3  4.114510e-08  1.090914e-07  1.841788e-07  3.080686e-07  2.402212e-07
4  3.987357e-09  1.373544e-08  2.634882e-08  4.449762e-08  3.527128e-08
```

```
   specz_redshift           g_r           r_i           i_y           y_z  \
0         0.31652 -2.239657e-08 -2.129643e-08 -3.000452e-08  1.380413e-08
1         0.56769 -4.345878e-09 -7.548391e-09 -8.218576e-09  3.129404e-09
2         0.53428 -1.224164e-08 -2.172218e-08 -2.761336e-08  9.193588e-09
3         0.11878 -6.794635e-08 -7.508739e-08 -1.238898e-07  6.784739e-08
4         0.23497 -9.748082e-09 -1.261338e-08 -1.814880e-08  9.226337e-09

   Cluster  cluster_dbscan  cluster_kmeans  cluster_gmm
0        2               0               3            3
1        1               0               1            1
2        1               0               3            3
3        2               0               2            2
4        1               0               3            3
Index(['g_flux', 'r_flux', 'i_flux', 'y_flux', 'z_flux', 'specz_redshift',
       'g_r', 'r_i', 'i_y', 'y_z', 'Cluster', 'cluster_dbscan',
       'cluster_kmeans', 'cluster_gmm'],
      dtype='object')
```

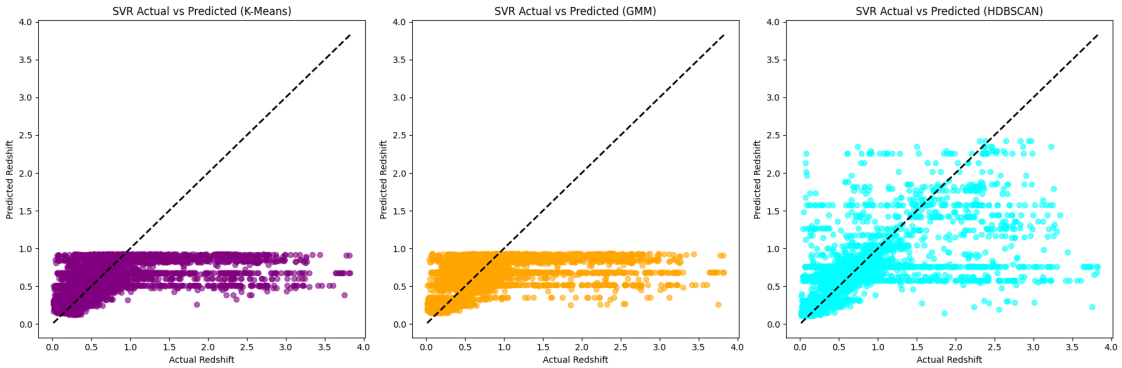
SVR and UMAP

[6]:
```python
# Add-Ons for Photometric Redshift Estimation
# Fixing UMAP Import Issue and Enhancing Preprocessing

# ---- STEP 11: Install and Import UMAP Properly ----
# Ensure proper UMAP installation: pip install umap-learn
from umap import UMAP

umap = UMAP(n_neighbors=15, min_dist=0.1, n_components=2, random_state=42)
X_umap = umap.fit_transform(df_transformed)

# ---- STEP 12: Add More Color Indices (Using Flux Values) ----
df_clean['u_g'] = df_clean['g_flux'] - df_clean['r_flux']
df_clean['g_r'] = df_clean['r_flux'] - df_clean['i_flux']
df_clean['r_i'] = df_clean['i_flux'] - df_clean['y_flux']
df_clean['i_z'] = df_clean['y_flux'] - df_clean['z_flux']

# ---- STEP 13: Additional Clustering (HDBSCAN) for SVR Comparison ----
try:
    from hdbscan import HDBSCAN
except ImportError:
    print("HDBSCAN not installed. Use: pip install hdbscan")
    raise

hdbscan = HDBSCAN(min_cluster_size=8)
df_clean['cluster_hdbscan'] = hdbscan.fit_predict(X_umap)

# ---- STEP 14: SVR Regression for Redshift Estimation ----
from sklearn.svm import SVR
```

```
svr_results = {}

for cluster_type in ['cluster_kmeans', 'cluster_gmm', 'cluster_hdbscan']:
    df_temp = pd.get_dummies(df_clean, columns=[cluster_type],␣
 ↪prefix=[f'clust_{cluster_type}'])
    X = df_temp.drop(columns=['specz_redshift'])
    y = df_temp['specz_redshift']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

    svr = SVR(kernel='rbf', C=1.0, epsilon=0.1)
    svr.fit(X_train, y_train)
    y_pred = svr.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    svr_results[cluster_type] = {'MSE': mse, 'R2': r2, 'y_pred': y_pred}

# ---- STEP 15: Plot SVR Results ----
fig, ax = plt.subplots(1, 3, figsize=(18, 6))
titles = ['K-Means', 'GMM', 'HDBSCAN']
colors = ['purple', 'orange', 'cyan']

for i, cluster_type in enumerate(svr_results.keys()):
    ax[i].scatter(y_test, svr_results[cluster_type]['y_pred'], alpha=0.6,␣
 ↪color=colors[i])
    ax[i].set_title(f'SVR Actual vs Predicted ({titles[i]})')
    ax[i].set_xlabel('Actual Redshift')
    ax[i].set_ylabel('Predicted Redshift')
    ax[i].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],␣
 ↪'k--', lw=2)

plt.tight_layout()
plt.show()

# ---- Print SVR Performance Metrics ----
print('SVR Regression Results:')
for cluster_type, metrics in svr_results.items():
    print(f"{cluster_type.upper()} - MSE: {metrics['MSE']:.6f}, R2:␣
 ↪{metrics['R2']:.6f}")
```

```
/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/deprecation.py:151: FutureWarning: 'force_all_finite' was
renamed to 'ensure_all_finite' in 1.6 and will be removed in 1.8.
  warnings.warn(
```

```
/home/chloy/miniconda3/lib/python3.10/site-packages/umap/umap_.py:1952:
UserWarning: n_jobs value 1 overridden to 1 by setting random_state. Use no seed
for parallelism.
  warn(
/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/deprecation.py:151: FutureWarning: 'force_all_finite' was
renamed to 'ensure_all_finite' in 1.6 and will be removed in 1.8.
  warnings.warn(
/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/deprecation.py:151: FutureWarning: 'force_all_finite' was
renamed to 'ensure_all_finite' in 1.6 and will be removed in 1.8.
  warnings.warn(
```




```
SVR Regression Results:
CLUSTER_KMEANS - MSE: 0.227899, R2: 0.281111
CLUSTER_GMM - MSE: 0.226127, R2: 0.286701
CLUSTER_HDBSCAN - MSE: 0.167378, R2: 0.472019
```

XGBoost , Gradient boosting and MLP NN

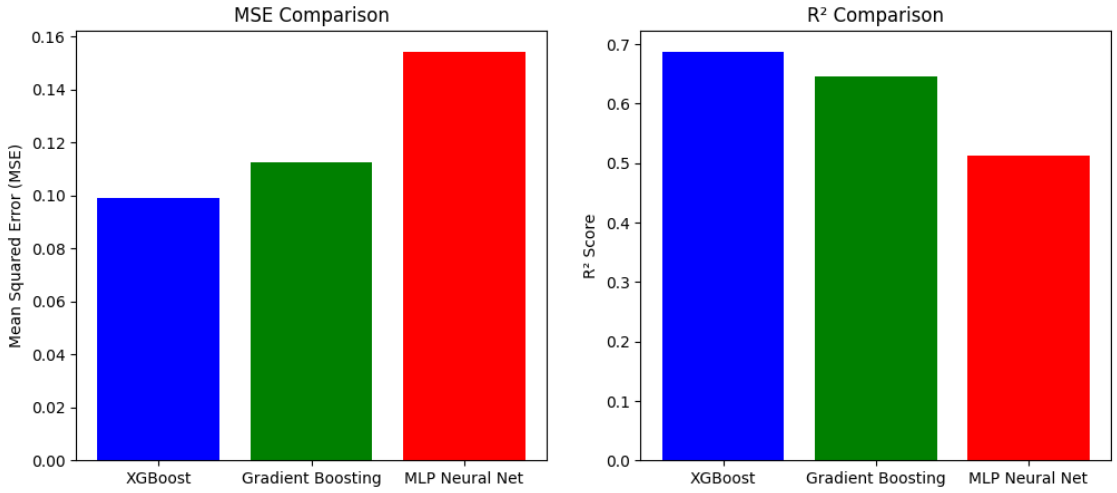
[7]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import GradientBoostingRegressor
from xgboost import XGBRegressor
from sklearn.neural_network import MLPRegressor
import faiss  # Faster nearest neighbor search

# ---- STEP 1: ADD MORE COLOR INDICES ----
# Verify flux columns exist before calculating new features
if all(col in df_clean.columns for col in ['g_flux', 'r_flux', 'i_flux',␣
 ↪'y_flux', 'z_flux']):
    df_clean['g_i'] = df_clean['g_flux'] - df_clean['i_flux']
```

```
    df_clean['r_z'] = df_clean['r_flux'] - df_clean['z_flux']
    df_clean['i_y_z'] = df_clean['i_flux'] - df_clean['y_flux'] -␣
 ↪df_clean['z_flux']
else:
    raise ValueError("Missing required flux columns in df_clean.")

# ---- STEP 2: DEFINE FEATURES & TARGET ----
features = ['g_flux', 'r_flux', 'i_flux', 'y_flux', 'z_flux',
            'g_r', 'r_i', 'i_y', 'y_z', 'g_i', 'r_z', 'i_y_z']

# Include cluster one-hot encoding for different clustering techniques
if 'cluster_kmeans' in df_clean.columns and 'cluster_gmm' in df_clean.columns␣
 ↪and 'cluster_hdbscan' in df_clean.columns:
    df_encoded = pd.get_dummies(df_clean, columns=['cluster_kmeans',␣
 ↪'cluster_gmm', 'cluster_hdbscan'],
                                prefix=['clust_kmeans', 'clust_gmm',␣
 ↪'clust_hdbscan'])
else:
    raise ValueError("Cluster columns are missing in df_clean.")

X = df_encoded[features + [col for col in df_encoded.columns if col.
 ↪startswith('clust_')]]
y = df_encoded['specz_redshift']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# ---- STEP 3: APPLY XGBOOST ----
xgb = XGBRegressor(n_estimators=200, learning_rate=0.1, random_state=42)
xgb.fit(X_train, y_train)
y_pred_xgb = xgb.predict(X_test)

mse_xgb = mean_squared_error(y_test, y_pred_xgb)
r2_xgb = r2_score(y_test, y_pred_xgb)

print(f"XGBoost - MSE: {mse_xgb:.6f}, R2: {r2_xgb:.6f}")

# ---- STEP 4: APPLY GRADIENT BOOSTING ----
gbr = GradientBoostingRegressor(n_estimators=150, learning_rate=0.05,␣
 ↪random_state=42)
gbr.fit(X_train, y_train)
y_pred_gbr = gbr.predict(X_test)

mse_gbr = mean_squared_error(y_test, y_pred_gbr)
r2_gbr = r2_score(y_test, y_pred_gbr)
```

```
print(f"Gradient Boosting - MSE: {mse_gbr:.6f}, R2: {r2_gbr:.6f}")

# ---- STEP 5: TRY MLP NEURAL NETWORK ----
mlp = MLPRegressor(hidden_layer_sizes=(64, 32), max_iter=500, random_state=42)
mlp.fit(X_train, y_train)
y_pred_mlp = mlp.predict(X_test)

mse_mlp = mean_squared_error(y_test, y_pred_mlp)
r2_mlp = r2_score(y_test, y_pred_mlp)

print(f"MLP Neural Net - MSE: {mse_mlp:.6f}, R2: {r2_mlp:.6f}")

# ---- STEP 6: OPTIONAL - FASTER NEAREST NEIGHBOR SEARCH FOR DBSCAN ----
index = faiss.IndexFlatL2(X_train.shape[1])  # L2 distance (Euclidean)
index.add(X_train.astype('float32'))  # FAISS requires float32
_, indices = index.search(X_train.astype('float32'), k=5)  # Find 5 nearest␣
 ↪neighbors
```

```
XGBoost - MSE: 0.099036, R2: 0.687598
Gradient Boosting - MSE: 0.112597, R2: 0.644823
MLP Neural Net - MSE: 0.154431, R2: 0.512862
```

[8]:
```
# ---- PLOT COMPARISON ----
models = ['XGBoost', 'Gradient Boosting', 'MLP Neural Net']
mse_scores = [mse_xgb, mse_gbr, mse_mlp]
r2_scores = [r2_xgb, r2_gbr, r2_mlp]

fig, ax = plt.subplots(1, 2, figsize=(12, 5))

# MSE Plot
ax[0].bar(models, mse_scores, color=['blue', 'green', 'red'])
ax[0].set_ylabel("Mean Squared Error (MSE)")
ax[0].set_title("MSE Comparison")

# R2 Score Plot
ax[1].bar(models, r2_scores, color=['blue', 'green', 'red'])
ax[1].set_ylabel("R² Score")
ax[1].set_title("R² Comparison")

plt.show()
```



Fine Tuning XGBoost Hyper-Parameters

[9]:
```
from xgboost import XGBRegressor
from sklearn.model_selection import GridSearchCV

# ---- XGBOOST HYPERPARAMETER TUNING ----
xgb_params = {
    'n_estimators': [200, 300, 500],  # More trees → better accuracy, but slower
    'learning_rate': [0.01, 0.05, 0.1],  # Lower = slower but more stable
    'max_depth': [5, 7, 9],  # Deeper trees can learn more but risk overfitting
    'subsample': [0.8, 1.0],  # Prevents overfitting by using a fraction of␣
 ↪data per tree
    'colsample_bytree': [0.8, 1.0]  # Randomly selects features per tree to␣
 ↪prevent overfitting
}

xgb = XGBRegressor(random_state=42)
xgb_grid = GridSearchCV(xgb, xgb_params, cv=3, scoring='r2', verbose=2,␣
 ↪n_jobs=-1)
xgb_grid.fit(X_train, y_train)

# Best model
best_xgb = xgb_grid.best_estimator_
y_pred_xgb = best_xgb.predict(X_test)

# Evaluate
mse_xgb = mean_squared_error(y_test, y_pred_xgb)
r2_xgb = r2_score(y_test, y_pred_xgb)

print(f"Tuned XGBoost - Best Params: {xgb_grid.best_params_}")
```

```
print(f"Tuned XGBoost - MSE: {mse_xgb:.6f}, R2: {r2_xgb:.6f}")
```

```
Fitting 3 folds for each of 108 candidates, totalling 324 fits
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=1.0; total time=  20.8s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=0.8; total time=  21.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=0.8; total time=  21.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=0.8; total time=  21.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=1.0; total time=  21.9s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=1.0; total time=  22.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=1.0; total time=  28.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=0.8; total time=  28.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=1.0; total time=  28.9s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=0.8; total time=  29.4s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=1.0; total time=  29.8s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=0.8; total time=  30.3s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=0.8; total time=  26.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=0.8; total time=  27.0s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=0.8; total time=  26.8s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=1.0; total time=  27.1s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=1.0; total time=  26.9s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=1.0; total time=  27.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=0.8; total time=  41.1s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=1.0; total time=  41.2s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=0.8; total time=  41.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=0.8; total time=  43.4s
```

```
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=1.0; total time=  42.0s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=1.0; total time=  43.0s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=0.8; total time=  39.3s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=0.8; total time=  39.3s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=0.8; total time=  40.2s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=1.0; total time=  39.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=1.0; total time=  39.4s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=1.0; total time=  39.4s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=0.8; total time=  54.8s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=0.8; total time=  54.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=0.8; total time=  56.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=1.0; total time=  56.3s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=1.0; total time=  57.2s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=1.0; total time=  58.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=0.8; total time=  35.9s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=0.8; total time=  36.4s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=0.8; total time=  36.7s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=1.0; total time=  36.9s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=1.0; total time=  37.5s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=1.0; total time=  37.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=0.8; total time=  50.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=0.8; total time=  50.5s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=0.8; total time=  52.6s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=1.0; total time=  52.3s
```

```
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=1.0; total time=  53.3s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=1.0; total time=  52.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=0.8; total time=  17.4s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=0.8; total time=  17.4s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=1.0; total time=  16.2s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=0.8; total time=  17.3s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=1.0; total time=  17.9s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=1.0; total time=  18.0s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=0.8; total time= 1.2min
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=0.8; total time= 1.2min
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=0.8; total time= 1.2min
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=0.8; total time=  21.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=0.8; total time=  22.5s
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=1.0; total time= 1.3min
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=1.0; total time= 1.3min
[CV] END colsample_bytree=0.8, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=1.0; total time= 1.3min
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=0.8; total time=  21.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=1.0; total time=  23.4s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=1.0; total time=  20.9s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=1.0; total time=  21.9s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=0.8; total time=  18.4s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=0.8; total time=  18.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=0.8; total time=  19.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=1.0; total time=  19.2s
```

```
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=1.0; total time=  20.1s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=1.0; total time=  19.0s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=0.8; total time=  32.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=0.8; total time=  33.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=0.8; total time=  33.4s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=1.0; total time=  32.0s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=1.0; total time=  34.2s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=1.0; total time=  34.1s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=0.8; total time=  25.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=0.8; total time=  24.6s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=0.8; total time=  26.1s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=1.0; total time=  25.6s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=1.0; total time=  25.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=1.0; total time=  26.6s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=0.8; total time=  37.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=0.8; total time=  38.4s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=0.8; total time=  37.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=0.8; total time=  24.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=0.8; total time=  23.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=1.0; total time=  38.7s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=1.0; total time=  37.3s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=0.8; total time=  22.9s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=1.0; total time=  38.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=1.0; total time=  24.1s
```

```
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=1.0; total time=  24.6s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=1.0; total time=  23.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=0.8; total time=  29.5s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=0.8; total time=  30.3s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=0.8; total time=  30.1s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=1.0; total time=  30.0s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=1.0; total time=  31.1s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=1.0; total time=  30.7s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=0.8; total time=  15.6s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=0.8; total time=  15.3s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=0.8; total time=  16.4s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=1.0; total time=  15.4s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=0.8; total time=  45.0s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=1.0; total time=  15.7s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=0.8; total time=  45.3s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=0.8; total time=  46.1s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=1.0; total time=  15.9s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=1.0; total time=  43.8s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=1.0; total time=  47.0s
[CV] END colsample_bytree=0.8, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=1.0; total time=  45.4s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=0.8; total time=  21.1s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=0.8; total time=  20.5s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=0.8; total time=  21.2s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=1.0; total time=  19.7s
```

```
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=1.0; total time=  20.8s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=1.0; total time=  21.4s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=0.8; total time=  31.9s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=0.8; total time=  33.0s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=0.8; total time=  32.5s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=0.8; total time=  17.0s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=0.8; total time=  17.5s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=0.8; total time=  17.8s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=1.0; total time=  17.5s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=1.0; total time=  33.2s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=1.0; total time=  32.0s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=1.0; total time=  33.7s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=1.0; total time=  17.2s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=1.0; total time=  17.4s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=0.8; total time=  22.7s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=0.8; total time=  22.5s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=1.0; total time=  23.4s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=1.0; total time=  22.9s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=1.0; total time=  23.5s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=0.8; total time=  27.7s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=0.8; total time=  35.8s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=0.8; total time=  36.6s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=0.8; total time=  36.1s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=1.0; total time=  37.0s
```

```
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=1.0; total time=  37.6s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=1.0; total time=  37.0s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=0.8; total time=  20.6s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=0.8; total time=  20.7s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=0.8; total time=  20.7s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=1.0; total time=  20.8s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=1.0; total time=  19.9s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=1.0; total time=  19.7s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=0.8; total time=  27.2s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=0.8; total time=  27.3s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=0.8; total time=  28.3s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=1.0; total time=  27.4s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=1.0; total time=  28.2s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=1.0; total time=  28.0s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=0.8; total time=  42.3s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=1.0; total time=  41.2s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=0.8; total time=  41.9s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=0.8; total time=  42.9s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=1.0; total time=  42.8s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=0.8; total time=  22.5s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=0.8; total time=  23.2s
[CV] END colsample_bytree=0.8, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=1.0; total time=  43.4s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=0.8; total time=  24.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=1.0; total time=  22.9s
```

```
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=1.0; total time=  23.0s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=200, subsample=1.0; total time=  22.8s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=0.8; total time=  32.0s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=0.8; total time=  33.0s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=1.0; total time=  32.3s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=0.8; total time=  32.7s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=1.0; total time=  30.8s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=300, subsample=1.0; total time=  31.3s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=0.8; total time=  43.5s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=0.8; total time=  43.7s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=0.8; total time=  45.2s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=1.0; total time=  44.4s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=1.0; total time=  43.9s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=5,
n_estimators=500, subsample=1.0; total time=  47.8s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=0.8; total time=  28.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=0.8; total time=  28.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=0.8; total time=  27.9s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=1.0; total time=  27.9s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=1.0; total time=  29.3s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=200, subsample=1.0; total time=  30.0s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=0.8; total time=  39.6s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=0.8; total time=  39.3s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=0.8; total time=  39.4s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=1.0; total time=  41.1s
```

```
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=1.0; total time=  41.4s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=300, subsample=1.0; total time=  41.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=0.8; total time=  53.6s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=0.8; total time=  54.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=0.8; total time=  55.3s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=1.0; total time=  57.8s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=1.0; total time=  59.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=0.8; total time=  37.6s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=0.8; total time=  36.7s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=0.8; total time=  38.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=7,
n_estimators=500, subsample=1.0; total time= 1.1min
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=1.0; total time=  37.9s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=1.0; total time=  39.6s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=200, subsample=1.0; total time=  39.4s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=0.8; total time=  51.8s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=0.8; total time=  53.2s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=0.8; total time=  52.9s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=1.0; total time=  57.1s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=1.0; total time=  57.2s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=300, subsample=1.0; total time=  55.6s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=0.8; total time=  17.8s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=0.8; total time=  17.2s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=0.8; total time=  16.8s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=1.0; total time=  17.4s
```

```
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=1.0; total time=  17.7s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=200, subsample=1.0; total time=  16.9s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=0.8; total time= 1.3min
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=0.8; total time= 1.3min
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=0.8; total time= 1.3min
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=0.8; total time=  21.7s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=0.8; total time=  22.2s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=0.8; total time=  22.5s
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=1.0; total time= 1.3min
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=1.0; total time= 1.4min
[CV] END colsample_bytree=1.0, learning_rate=0.01, max_depth=9,
n_estimators=500, subsample=1.0; total time= 1.3min
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=1.0; total time=  22.4s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=1.0; total time=  22.5s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=300, subsample=1.0; total time=  21.0s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=0.8; total time=  20.0s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=0.8; total time=  19.4s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=0.8; total time=  19.0s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=0.8; total time=  34.1s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=0.8; total time=  32.6s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=1.0; total time=  20.0s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=0.8; total time=  33.7s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=1.0; total time=  32.8s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=1.0; total time=  19.8s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=200, subsample=1.0; total time=  19.9s
```

```
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=1.0; total time=  34.1s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=5,
n_estimators=500, subsample=1.0; total time=  33.8s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=0.8; total time=  25.9s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=0.8; total time=  25.7s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=0.8; total time=  26.0s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=1.0; total time=  26.1s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=1.0; total time=  27.6s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=300, subsample=1.0; total time=  26.3s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=0.8; total time=  38.4s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=0.8; total time=  38.4s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=1.0; total time=  38.5s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=1.0; total time=  39.7s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=0.8; total time=  23.6s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=1.0; total time=  39.1s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=7,
n_estimators=500, subsample=0.8; total time=  42.7s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=0.8; total time=  24.0s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=0.8; total time=  23.3s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=1.0; total time=  24.9s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=1.0; total time=  24.4s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=200, subsample=1.0; total time=  25.2s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=0.8; total time=  30.7s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=0.8; total time=  31.2s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=0.8; total time=  31.2s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=1.0; total time=  32.4s
```

```
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=1.0; total time=  31.4s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=300, subsample=1.0; total time=  32.3s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=0.8; total time=  15.4s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=0.8; total time=  15.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=1.0; total time=  15.3s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=0.8; total time=  44.9s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=1.0; total time=  15.1s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=0.8; total time=  19.2s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=200,
subsample=1.0; total time=  15.8s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=0.8; total time=  45.6s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=0.8; total time=  47.0s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=1.0; total time=  45.8s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=1.0; total time=  45.2s
[CV] END colsample_bytree=1.0, learning_rate=0.05, max_depth=9,
n_estimators=500, subsample=1.0; total time=  45.7s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=0.8; total time=  21.5s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=0.8; total time=  20.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=0.8; total time=  20.3s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=1.0; total time=  20.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=1.0; total time=  22.0s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=300,
subsample=1.0; total time=  21.3s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=0.8; total time=  32.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=0.8; total time=  17.2s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=0.8; total time=  32.5s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=0.8; total time=  17.1s
```

```
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=0.8; total time=  32.7s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=0.8; total time=  17.2s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=1.0; total time=  17.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=1.0; total time=  18.0s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=1.0; total time=  33.9s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=1.0; total time=  34.0s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=5, n_estimators=500,
subsample=1.0; total time=  33.4s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=200,
subsample=1.0; total time=  18.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=0.8; total time=  23.8s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=0.8; total time=  23.9s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=1.0; total time=  23.3s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=0.8; total time=  24.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=1.0; total time=  24.2s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=300,
subsample=1.0; total time=  24.3s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=0.8; total time=  36.9s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=1.0; total time=  37.5s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=0.8; total time=  38.1s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=0.8; total time=  37.8s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=1.0; total time=  38.1s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=0.8; total time=  20.5s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=7, n_estimators=500,
subsample=1.0; total time=  38.7s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=0.8; total time=  20.9s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=1.0; total time=  20.4s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=0.8; total time=  21.4s
```

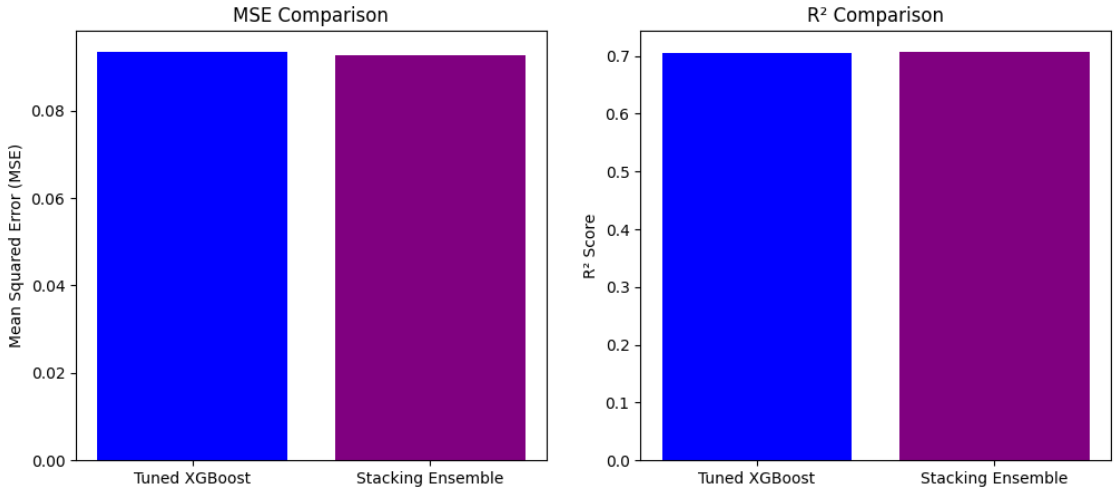
```
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=1.0; total time=  21.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=200,
subsample=1.0; total time=  20.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=0.8; total time=  27.1s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=0.8; total time=  27.4s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=0.8; total time=  27.9s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=1.0; total time=  27.8s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=1.0; total time=  26.8s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=300,
subsample=1.0; total time=  27.4s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=0.8; total time=  34.6s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=0.8; total time=  33.5s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=0.8; total time=  34.4s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=1.0; total time=  33.4s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=1.0; total time=  32.5s
[CV] END colsample_bytree=1.0, learning_rate=0.1, max_depth=9, n_estimators=500,
subsample=1.0; total time=  32.9s
Tuned XGBoost - Best Params: {'colsample_bytree': 0.8, 'learning_rate': 0.01,
'max_depth': 9, 'n_estimators': 500, 'subsample': 0.8}
Tuned XGBoost - MSE: 0.093542, R2: 0.704931
```

Emsemble (XGBoost + RF + Gradient Boost)

[10]:
```python
from sklearn.ensemble import StackingRegressor, RandomForestRegressor,␣
 ↪GradientBoostingRegressor
from sklearn.linear_model import Ridge  # Final meta-learner

# Define base models
base_models = [
    ('xgb', best_xgb),  # Best tuned XGBoost
    ('rf', RandomForestRegressor(n_estimators=200, random_state=42)),
    ('gbr', GradientBoostingRegressor(n_estimators=150, learning_rate=0.05,␣
 ↪random_state=42))
]

# Stacking Ensemble
```

```
stacking_model = StackingRegressor(estimators=base_models,␣
 ↪final_estimator=Ridge())

# Train Stacking Model
stacking_model.fit(X_train, y_train)
y_pred_stack = stacking_model.predict(X_test)

# Evaluate
mse_stack = mean_squared_error(y_test, y_pred_stack)
r2_stack = r2_score(y_test, y_pred_stack)

print(f"Stacking Ensemble - MSE: {mse_stack:.6f}, R2: {r2_stack:.6f}")


# ---- PLOT COMPARISON ----
models = ['Tuned XGBoost', 'Stacking Ensemble']
mse_scores = [mse_xgb, mse_stack]
r2_scores = [r2_xgb, r2_stack]

fig, ax = plt.subplots(1, 2, figsize=(12, 5))

# MSE Plot
ax[0].bar(models, mse_scores, color=['blue', 'purple'])
ax[0].set_ylabel("Mean Squared Error (MSE)")
ax[0].set_title("MSE Comparison")

# R2 Score Plot
ax[1].bar(models, r2_scores, color=['blue', 'purple'])
ax[1].set_ylabel("R² Score")
ax[1].set_title("R² Comparison")

plt.show()
```

```
Stacking Ensemble - MSE: 0.092646, R2: 0.707756
```



Spectral Clustering

[11]:
```
from sklearn.cluster import SpectralClustering
import pandas as pd
import numpy as np

# ---- USE OPTIMAL K FROM K-MEANS ----
spectral = SpectralClustering(n_clusters=optimal_k,␣
 ↪affinity='nearest_neighbors', random_state=42)
df_clean['cluster_spectral'] = spectral.fit_predict(X)  #  Spectral Clustering␣
 ↪Labels

# ---- ONE-HOT ENCODE CLUSTERING ----
df_encoded = pd.get_dummies(df_clean, columns=['cluster_kmeans', 'cluster_gmm',␣
 ↪'cluster_spectral'],
                            prefix=['clust_kmeans', 'clust_gmm',␣
 ↪'clust_spectral'])

print("Clustering features successfully added!")
```

```
/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/manifold/_spectral_embedding.py:329: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
  warnings.warn(
```

```
Clustering features successfully added!
```

Ensemble Stacking Model (XGBoost+RF+Gradient Boost +LightGBM)

[12]:
```
from sklearn.ensemble import StackingRegressor, RandomForestRegressor,␣
 ↪GradientBoostingRegressor
```

```
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# ---- DEFINE FEATURES & TARGET ----
X = df_encoded.drop(columns=['specz_redshift'])  #  Flux features are still␣
 ↪here!
y = df_encoded['specz_redshift']

# ---- TRAIN-TEST SPLIT ----
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# ---- DEFINE BASE MODELS ----
base_models = [
    ('xgb', XGBRegressor(n_estimators=300, learning_rate=0.05, max_depth=7,␣
 ↪random_state=42)),
    ('rf', RandomForestRegressor(n_estimators=200, random_state=42)),
    ('gbr', GradientBoostingRegressor(n_estimators=150, learning_rate=0.05,␣
 ↪random_state=42)),
    ('lgbm', LGBMRegressor(n_estimators=200, learning_rate=0.05,␣
 ↪random_state=42))
]

# ---- USE LightGBM AS META-LEARNER ----
stacking_model = StackingRegressor(estimators=base_models,␣
 ↪final_estimator=LGBMRegressor())

# ---- TRAIN STACKING REGRESSOR ----
stacking_model.fit(X_train, y_train)
y_pred_stack = stacking_model.predict(X_test)

# ---- EVALUATE PERFORMANCE ----
mse_stack = mean_squared_error(y_test, y_pred_stack)
r2_stack = r2_score(y_test, y_pred_stack)

print(f" Optimized Stacking Ensemble (K-Means + GMM + Spectral) - MSE:␣
 ↪{mse_stack:.6f}, R2: {r2_stack:.6f}")
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002285 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
[LightGBM] [Info] Number of data points in the train set: 32076, number of used
features: 29
[LightGBM] [Info] Start training from score 0.592784
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001999 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 25660, number of used
features: 29
[LightGBM] [Info] Start training from score 0.590185
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002036 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.593904
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001558 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.591059
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001054 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.595071
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001042 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.593702
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000381 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 32076, number of used
features: 4
[LightGBM] [Info] Start training from score 0.592784
  Optimized Stacking Ensemble (K-Means + GMM + Spectral) - MSE: 0.097463, R2:
0.692560

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
```

```
  warnings.warn(
```

RandomizedSearchCV for best hyperparameters For Fine Tuned Ensemble Stcked model

[13]:
```
from sklearn.model_selection import RandomizedSearchCV
import numpy as np
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor,␣
 ↪StackingRegressor

# ---- OPTIMIZED HYPERPARAMETER GRID ----
param_dist = {
    'xgb__n_estimators': np.arange(250, 400, 50),  #  Reduced range for␣
 ↪efficiency
    'xgb__learning_rate': [0.03, 0.05, 0.07],  #  Slightly reduced values
    'xgb__max_depth': [5, 7],
    'rf__n_estimators': [150, 200, 250],
    'gbr__n_estimators': [100, 150, 200],
    'gbr__learning_rate': [0.03, 0.05],
    'lgbm__n_estimators': [150, 200, 250],
    'lgbm__learning_rate': [0.03, 0.05]
}

# ---- DEFINE BASE MODELS ----
base_models = [
    ('xgb', XGBRegressor(n_estimators=300, learning_rate=0.05, max_depth=7,␣
 ↪random_state=42)),
    ('rf', RandomForestRegressor(n_estimators=200, random_state=42)),
    ('gbr', GradientBoostingRegressor(n_estimators=150, learning_rate=0.05,␣
 ↪random_state=42)),
    ('lgbm', LGBMRegressor(n_estimators=200, learning_rate=0.05,␣
 ↪random_state=42))
]

# ---- STACKING REGRESSOR WITH LightGBM META-LEARNER ----
stacking_model = StackingRegressor(estimators=base_models,␣
 ↪final_estimator=LGBMRegressor())

# ---- FAST RANDOMIZED SEARCH ----
random_search = RandomizedSearchCV(
    stacking_model, param_dist,
    n_iter=20,  #  Instead of testing 5000+ models, it tests only 20 random␣
 ↪configurations
    cv=2, scoring='r2', n_jobs=-1, verbose=2, random_state=42
)
random_search.fit(X_train, y_train)
```

```
# ---- PRINT BEST PARAMETERS ----
print("Best Parameters for Stacking:", random_search.best_params_)

# ---- TRAIN FINAL MODEL WITH BEST PARAMETERS ----
best_model = random_search.best_estimator_
y_pred_stack = best_model.predict(X_test)

# ---- EVALUATE PERFORMANCE ----
mse_stack = mean_squared_error(y_test, y_pred_stack)
r2_stack = r2_score(y_test, y_pred_stack)

print(f" Fine-Tuned Stacking Ensemble (Optimized) - MSE: {mse_stack:.6f}, R2:␣
 ↪{r2_stack:.6f}")
```

```
Fitting 2 folds for each of 20 candidates, totalling 40 fits
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.010383 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003815 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.008002 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013556 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.007243 seconds.
```

```
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006748 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005660 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.009024 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004236 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006873 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002831 seconds.
You can set `force_row_wise=true` to remove the overhead.
```

```
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005120 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004183 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003262 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003687 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.017816 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.007269 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
```

```
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008652 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004351 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001721 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.019584 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008570 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002271 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
```

```
/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=5, xgb__n_estimators=250; total
time=10.6min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004089 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=5, xgb__n_estimators=250; total
time=10.6min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.012539 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002717 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009244 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
```

```
testing was 0.003164 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006796 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.018256 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.009435 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.007454 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004690 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002006 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006475 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=7, xgb__n_estimators=300; total
time=12.2min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000684 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=7, xgb__n_estimators=300; total
time=12.3min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006339 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004300 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
```

```
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.024541 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.008535 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.010542 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001708 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006844 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.008999 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
```

```
testing was 0.012553 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005537 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002357 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.018427 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003897 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000690 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001998 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.595743

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.03, xgb__max_depth=5, xgb__n_estimators=300; total
time=15.0min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000600 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.03, xgb__max_depth=5, xgb__n_estimators=300; total
time=15.1min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003516 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002142 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.007818 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
```

```
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.014411 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005013 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000896 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003964 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=350; total
time=15.5min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013308 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008679 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.020418 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003452 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003618 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.008948 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009872 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
```

```
testing was 0.001764 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010405 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.012957 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005683 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006143 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001217 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001632 seconds.
You can set `force_row_wise=true` to remove the overhead.
```

```
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004123 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=350; total
time=16.3min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001012 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000739 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=250,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=300; total
time=16.4min

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
```

```
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=250,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=300; total
time=16.4min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.007599 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013105 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.005550 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013468 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000416 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009103 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009399 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002658 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002713 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008407 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002031 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.007927 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
```

```
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=350; total
time=17.0min

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=350; total
time=17.0min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.014918 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.007308 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.014337 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.012298 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.012321 seconds.
You can set `force_col_wise=true` to remove the overhead.
```

```
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.014306 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.012281 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002286 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003771 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001431 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001841 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
```

```
testing was 0.001784 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010417 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001188 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004958 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=150, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=7, xgb__n_estimators=350; total
time=11.0min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003392 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003584 seconds.
You can set `force_row_wise=true` to remove the overhead.
```

```
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010402 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003278 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003639 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=150, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=7, xgb__n_estimators=350; total
time=11.4min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.007555 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.005095 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
```

```
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004926 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.014059 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.017638 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002101 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.017610 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004275 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010806 seconds.
```

```
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001104 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.019366 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=7, xgb__n_estimators=300; total
time=11.2min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000627 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=7, xgb__n_estimators=300; total
time=11.4min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.008216 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
```

```
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.015358 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000972 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.007021 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010291 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.011398 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004286 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
```

```
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003251 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003386 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000327 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005410 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=350; total
time=12.9min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.005540 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
```

```
testing was 0.003667 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003881 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002122 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002146 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=350; total
time=12.6min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013935 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006489 seconds.
```

```
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.008806 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010390 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010314 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013372 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.005025 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004824 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
```

```
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006549 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002763 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002292 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.05, lgbm__n_estimators=150, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=7, xgb__n_estimators=350; total
time=14.8min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000743 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.05, lgbm__n_estimators=150, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=7, xgb__n_estimators=350; total
time=14.8min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
```

```
testing was 0.010336 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004652 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004562 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004303 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004857 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002663 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003500 seconds.
You can set `force_row_wise=true` to remove the overhead.
```

```
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.007992 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001297 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005371 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000836 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.010875 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003970 seconds.
You can set `force_row_wise=true` to remove the overhead.
```

```
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.005132 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=150, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=300; total
time=15.7min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013019 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008636 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005822 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(
```

```
[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=150, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=300; total
time=15.8min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.015253 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006160 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.016595 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004304 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002609 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005252 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000912 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006257 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002158 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=250; total
time=15.7min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008235 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002267 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
```

```
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=250; total
time=15.9min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.015364 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.015759 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.016326 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009909 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009650 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008571 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009713 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008830 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009906 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006520 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009421 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006532 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.007264 seconds.
```

```
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004633 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002213 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.009385 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004225 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.005312 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008563 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
```

```
/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=150, rf__n_estimators=250,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=350; total
time=16.2min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.011301 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.019499 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001877 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.011343 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003564 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386

/home/chloy/miniconda3/lib/python3.10/site-
```

```
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=350; total
time=14.5min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006547 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006294 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001931 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000299 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=150, rf__n_estimators=250,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=350; total
time=16.3min

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
```

```
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=350; total
time=14.6min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010283 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010210 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.007628 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.006693 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001662 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.018784 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.005045 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.006720 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 29
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001260 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001923 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=350; total
time=12.4min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002928 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
```

```
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004939 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.003709 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.013716 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001235 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005343 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.011895 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
```

```
testing was 0.005668 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004405 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.013663 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.008863 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002653 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.002764 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
```

```
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=350; total
time=12.8min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003291 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=7, xgb__n_estimators=250; total
time=11.9min
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001297 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.004785 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001415 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
```

```
testing was 0.000547 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.012321 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=200,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.03, xgb__max_depth=7, xgb__n_estimators=250; total
time=11.9min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001391 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004404 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001157 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001283 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000613 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=150,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=250; total time=
9.7min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.012304 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001383 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001223 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001595 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
```

```
testing was 0.001007 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000552 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=200, rf__n_estimators=150,
xgb__learning_rate=0.07, xgb__max_depth=7, xgb__n_estimators=250; total time=
9.9min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000935 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001706 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001381 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001442 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001514 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000634 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.03, xgb__max_depth=5, xgb__n_estimators=350; total
time=13.3min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001189 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.005024 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002797 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
```

```
testing was 0.003988 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001776 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003457 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003757 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001564 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.002700 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001467 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
```

```
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001473 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.003668 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.010355 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.004898 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=250; total time=
8.5min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000593 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001369 seconds.
```

```
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001176 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000519 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.05, gbr__n_estimators=150,
lgbm__learning_rate=0.05, lgbm__n_estimators=250, rf__n_estimators=250,
xgb__learning_rate=0.03, xgb__max_depth=5, xgb__n_estimators=350; total
time=13.5min

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=150,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=150,
xgb__learning_rate=0.05, xgb__max_depth=5, xgb__n_estimators=250; total time=
8.4min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000943 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3850
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598367
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001157 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
```

```
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.593794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001032 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.601214
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001066 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.598386
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001063 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586374
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001415 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3849
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.595743
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000307 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.597501
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001156 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.590235
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001129 seconds.
```

```
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3847
[LightGBM] [Info] Number of data points in the train set: 12830, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588099
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001133 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.586652
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001159 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3848
[LightGBM] [Info] Number of data points in the train set: 12831, number of used
features: 28
[LightGBM] [Info] Start training from score 0.588979
[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=7, xgb__n_estimators=250; total time=
9.1min

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000560 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 16038, number of used
features: 4
[LightGBM] [Info] Start training from score 0.588068

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(

[CV] END gbr__learning_rate=0.03, gbr__n_estimators=100,
lgbm__learning_rate=0.03, lgbm__n_estimators=250, rf__n_estimators=200,
xgb__learning_rate=0.05, xgb__max_depth=7, xgb__n_estimators=250; total time=
9.1min
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001919 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
```

```
[LightGBM] [Info] Number of data points in the train set: 32076, number of used
features: 29
[LightGBM] [Info] Start training from score 0.592784
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001634 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 25660, number of used
features: 29
[LightGBM] [Info] Start training from score 0.590185
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001771 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.593904
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001690 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.591059
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000902 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3852
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.595071
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.001781 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 3853
[LightGBM] [Info] Number of data points in the train set: 25661, number of used
features: 29
[LightGBM] [Info] Start training from score 0.593702
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000286 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1020
[LightGBM] [Info] Number of data points in the train set: 32076, number of used
features: 4
[LightGBM] [Info] Start training from score 0.592784
Best Parameters for Stacking: {'xgb__n_estimators': 300, 'xgb__max_depth': 7,
'xgb__learning_rate': 0.05, 'rf__n_estimators': 200, 'lgbm__n_estimators': 200,
'lgbm__learning_rate': 0.03, 'gbr__n_estimators': 100, 'gbr__learning_rate':
```

```
0.05}
  Fine-Tuned Stacking Ensemble (Optimized) - MSE: 0.097080, R2: 0.693769

/home/chloy/miniconda3/lib/python3.10/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LGBMRegressor was fitted with feature names
  warnings.warn(
```