Import required packages

In [169]:

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split, RandomizedSearchCV, GridSearchCV,
cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.decomposition import PCA
from sklearn.metrics import accuracy_score, plot_confusion_matrix, classification_report
from sklearn.feature_selection import SelectFromModel
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier

from skopt import BayesSearchCV
from skopt.space import Integer, Categorical, Real

import xgboost as xgb
import lightgbm as lgb
import catboost as ctb

from mlxtend.classifier import StackingClassifier, StackingCVClassifier

from pprint import pprint
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import seaborn as sns

RANDOM_SEED = 33
```

1. Load data and show the summary statistics

In [45]:

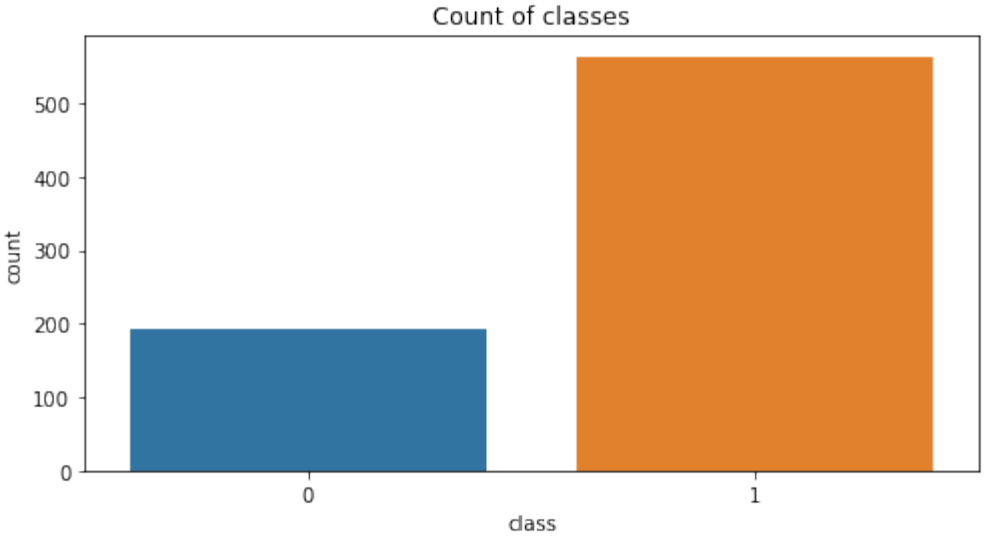
```
data = pd.read_csv(
    'dataset/parkinsons.csv')  # Load data and convert to DataFrame object

X = data.drop(['id', 'class'], axis=1)  # extracts feature data from the data
y = data['class'].to_numpy(
    np.int32)  # extracts labels for each samples and covert to an array

FEATURES = X.columns.to_numpy(
)  # extracts feature names and convert to an array
```

In [46]:

```
labels, label_counts = np.unique(
    y, return_counts=True)  # counts the number of each label

for i in range(len(labels)):
```

```
    print('Number of samples labelled to {}: {}'.format(
        labels[i], label_counts[i]))

print('Number of total samples: {}'.format(sum(label_counts)))
print('Number of total features: {}'.format(len(FEATURES)))

plt.figure(figsize=(8, 4))
sns.countplot(x='class', data=data)
plt.title('Count of classes')
plt.show()
```

```
Number of samples labelled to 0: 192
Number of samples labelled to 1: 564
Number of total samples: 756
Number of total features: 753
```



In [47]:

```
X.describe()  # display the summary stacts of the dataset
```

Out[47]:

	gender	PPE	DFA	RPDE	numPulses	numPeriodsPulses	meanPeriodPulses	stdDevPeriodPulses	locP
count	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000	756.
mean	0.515873	0.746284	0.700414	0.489058	323.972222	322.678571	0.006360	0.000383	0.
std	0.500079	0.169294	0.069718	0.137442	99.219059	99.402499	0.001826	0.000728	0.
min	0.000000	0.041551	0.543500	0.154300	2.000000	1.000000	0.002107	0.000011	0
25%	0.000000	0.762833	0.647053	0.386537	251.000000	250.000000	0.005003	0.000049	0.
50%	1.000000	0.809655	0.700525	0.484355	317.000000	316.000000	0.006048	0.000077	0.
75%	1.000000	0.834315	0.754985	0.586515	384.250000	383.250000	0.007528	0.000171	0.
max	1.000000	0.907660	0.852640	0.871230	907.000000	905.000000	0.012966	0.003483	0.

8 rows × 753 columns

In [48]:

```
data.drop(['id'], axis=1).head()  # Show the details of first 5 samples
```

Out[48]:

	gender	PPE	DFA	RPDE	numPulses	numPeriodsPulses	meanPeriodPulses	stdDevPeriodPulses	locPctJitter	locAbsJitt
0	1	0.85247	0.71826	0.57227	240	239	0.008064	0.000087	0.00218	0.0000
1	1	0.76686	0.69481	0.53966	234	233	0.008258	0.000073	0.00195	0.0000
2	1	0.85083	0.67604	0.58982	232	231	0.008340	0.000060	0.00176	0.0000
3	0	0.41121	0.79672	0.59257	178	177	0.010858	0.000183	0.00419	0.0000
4	0	0.32790	0.79782	0.53028	236	235	0.008162	0.002669	0.00535	0.0000

5 rows × 754 columns

3. Split samples to training dataset and test dataset

In [49]:

```
# Split samples to training data (80%) and test data(20%)
X_trn, X_tst, y_train, y_test = train_test_split(X,
                                                 y,
                                                 test_size=0.2,
                                                 random_state=RANDOM_SEED)
```

3. Data Preprocessing

Transform data to usable and meaningful information using sklearn.preprocessing.StandardScaler

$$x' = \frac{x - \bar{X}}{S}$$

In [50]:

```
## Standardize features
scaler = StandardScaler().fit(X_trn)
X_train = scaler.transform(X_trn)
X_test = scaler.transform(X_tst)
```

4. Feature Selection

1. Calculate the feature weights with sklearn.ensemble.RandomForestClassifier
2. Remove unrelated/unimportant feature with sklearn.feature_selection.SelectFromModel

4.1 Train the data with random forest classifier

In [51]:

```
clf = RandomForestClassifier(n_estimators=200,
                             random_state=RANDOM_SEED,
                             n_jobs=-1)
clf.fit(X_train, y_train)
```

Out[51]:

```
RandomForestClassifier(n_estimators=200, n_jobs=-1, random_state=33)
```

In [52]:

```
# print the feature name and corresponding feature importance
for feature in zip(FEATURES, clf.feature_importances_):
```

```
    print(feature)
```

```
('gender', 0.0)
('PPE', 0.001389557807225499)
('DFA', 0.0016029864966852008)
('RPDE', 0.000964521417603594)
('numPulses', 0.0008437166467154409)
('numPeriodsPulses', 0.00047039827208804764)
('meanPeriodPulses', 0.000949462056254064)
('stdDevPeriodPulses', 0.0012811209536960368)
('locPctJitter', 0.0016287128689146688)
('locAbsJitter', 0.0026162067755005403)
('rapJitter', 0.0015566274085830366)
('ppq5Jitter', 0.0008834034763984354)
('ddpJitter', 0.0023755471269437593)
('locShimmer', 0.0010495296958399174)
('locDbShimmer', 0.0022618936236616233)
('apq3Shimmer', 0.000384079311034623)
('apq5Shimmer', 0.0005451904355748672)
('apq11Shimmer', 0.0021426199339519753)
('ddaShimmer', 0.0009713534211062107)
('meanAutoCorrHarmonicity', 0.00042786272201246164)
('meanNoiseToHarmHarmonicity', 0.0012715842120737122)
('meanHarmToNoiseHarmonicity', 0.0009894894026296762)
('minIntensity', 0.0020281691545168414)
('maxIntensity', 0.001152874776565143)
('meanIntensity', 0.0015044953973579328)
('f1', 0.0016714176720634669)
('f2', 0.0010502751377782485)
('f3', 0.0007722277090097884)
('f4', 0.0007613498997088789)
('b1', 0.00099337770584766)
('b2', 0.0006694181877162475)
('b3', 0.00042823999810547735)
('b4', 0.0006290052800578259)
('GQ_prc5_95', 0.000853964199901905)
('GQ_std_cycle_open', 0.004508411002139156)
('GQ_std_cycle_closed', 0.0003395286084094701)
('GNE_mean', 0.0008302828857221372)
('GNE_std', 0.0018211201827412979)
('GNE_SNR_TKEO', 0.00157043528668777)
('GNE_SNR_SEO', 0.0004871081585851354 6)
('GNE_NSR_TKEO', 0.002310565976268404)
('GNE_NSR_SEO', 0.0015439186656187875)
('VFER_mean', 0.0006220898244310252)
('VFER_std', 0.0004001334752299669)
('VFER_entropy', 0.0006471755384372274)
('VFER_SNR_TKEO', 0.0008406293778885786)
('VFER_SNR_SEO', 0.0007982465335436871)
('VFER_NSR_TKEO', 0.00037734177205468494)
('VFER_NSR_SEO', 0.0005435237756876005)
('IMF_SNR_SEO', 0.0008465773743943367)
('IMF_SNR_TKEO', 0.0007046749881335769)
('IMF_SNR_entropy', 0.0010099547797461667)
('IMF_NSR_SEO', 0.0005503074031230228)
('IMF_NSR_TKEO', 0.0003115570890499211)
('IMF_NSR_entropy', 0.0004991323430404769)
('mean_Log_energy', 0.0011508627859303795)
('mean_MFCC_0th_coef', 0.0016648952556295156)
('mean_MFCC_1st_coef', 0.0006129797998405651)
('mean_MFCC_2nd_coef', 0.00980645154423479)
('mean_MFCC_3rd_coef', 0.0018044195231143474)
('mean_MFCC_4th_coef', 0.0009885488261669264)
('mean_MFCC_5th_coef', 0.0015877408133488466)
('mean_MFCC_6th_coef', 0.001862501279502293)
('mean_MFCC_7th_coef', 0.0008305021621621041)
('mean_MFCC_8th_coef', 0.0010849258335476504)
('mean_MFCC_9th_coef', 0.0007850349932085834)
('mean_MFCC_10th_coef', 0.002040628922443932)
```

```
('mean_MFCC_11th_coef', 0.0006222099974290991)
('mean_MFCC_12th_coef', 0.0006835112301116322)
('mean_delta_log_energy', 0.00059314153935458 3)
('mean_0th_delta', 0.0016257118999126357)
('mean_1st_delta', 0.0008087036441797407)
('mean_2nd_delta', 0.0007457159993540553)
('mean_3rd_delta', 0.0008815295045472967)
('mean_4th_delta', 0.0005509987805636102)
('mean_5th_delta', 0.0013618596211092651)
('mean_6th_delta', 0.0009710712758108435)
('mean_7th_delta', 0.000693241406458633)
('mean_8th_delta', 0.0004203378977281 0703)
('mean_9th_delta', 0.0005794236495254841)
('mean_10th_delta', 0.0015684919554393361)
('mean_11th_delta', 0.0006093559611652626)
('mean_12th_delta', 0.0007734888230028306)
('mean_delta_delta_log_energy', 0.000774156824855549)
('mean_delta_delta_0th', 0.00021144836932666724)
('mean_1st_delta_delta', 0.0004654274682237 9185)
('mean_2nd_delta_delta', 0.0004878451998657678)
('mean_3rd_delta_delta', 0.00027968212988362107)
('mean_4th_delta_delta', 0.0005029530024988305)
('mean_5th_delta_delta', 0.00024792149255307674)
('mean_6th_delta_delta', 0.0012950156061263887)
('mean_7th_delta_delta', 0.0012351502993240865)
('mean_8th_delta_delta', 0.0007961580975282919)
('mean_9th_delta_delta', 0.0004291700187908 5693)
('mean_10th_delta_delta', 0.0009033957028916739)
('mean_11th_delta_delta', 0.0003322976871917487)
('mean_12th_delta_delta', 0.0008952551493054293)
('std_Log_energy', 0.003958370118004985)
('std_MFCC_0th_coef', 0.0005551734708968446)
('std_MFCC_1st_coef', 0.00148174126525251)
('std_MFCC_2nd_coef', 0.0007146992906468319)
('std_MFCC_3rd_coef', 0.0008558894185930348)
('std_MFCC_4th_coef', 0.0010309564293618095)
('std_MFCC_5th_coef', 0.0005934435512543465)
('std_MFCC_6th_coef', 0.0009557792281041832)
('std_MFCC_7th_coef', 0.002157633607124108)
('std_MFCC_8th_coef', 0.001579558844955438)
('std_MFCC_9th_coef', 0.0010859532579906314)
('std_MFCC_10th_coef', 0.00106866683860128)
('std_MFCC_11th_coef', 0.0003840826321856343)
('std_MFCC_12th_coef', 0.0013257784217112518)
('std_delta_log_energy', 0.011432374745710405)
('std_0th_delta', 0.0012028923730789624)
('std_1st_delta', 0.00129180431670027)
('std_2nd_delta', 0.0017628742516945268)
('std_3rd_delta', 0.0012679608462274124)
('std_4th_delta', 0.0017851456224751466)
('std_5th_delta', 0.0011956881256254132)
('std_6th_delta', 0.005758804255876965)
('std_7th_delta', 0.004364044763545143)
('std_8th_delta', 0.006675134746688542)
('std_9th_delta', 0.006010709628964289)
('std_10th_delta', 0.004319787127420889)
('std_11th_delta', 0.0027682077475129395)
('std_12th_delta', 0.0019213948882096038)
('std_delta_delta_log_energy', 0.014350356148289207)
('std_delta_delta_0th', 0.0009968970184191708)
('std_1st_delta_delta', 0.0009536047764437367)
('std_2nd_delta_delta', 0.00108595547083765)
('std_3rd_delta_delta', 0.0014972870182555706)
('std_4th_delta_delta', 0.0011398558491178398)
('std_5th_delta_delta', 0.001600870519047812)
('std_6th_delta_delta', 0.00377615456604098)
('std_7th_delta_delta', 0.006881323075179752)
('std_8th_delta_delta', 0.004842886563100295)
('std_9th_delta_delta', 0.003553562167309921)
```

```
('std_10th_delta_delta', 0.002837200512303935)
('std_11th_delta_delta', 0.0012010219722672888)
('std_12th_delta_delta', 0.003149787735665546)
('Ea', 0.00020855076423133905)
('Ed_1_coef', 0.001376567818186888)
('Ed_2_coef', 0.0005702692095536425)
('Ed_3_coef', 0.0006602997885964236)
('Ed_4_coef', 0.0016378965165866086)
('Ed_5_coef', 0.0006023399862570446)
('Ed_6_coef', 0.000976662288539203)
('Ed_7_coef', 0.0007665615560772241)
('Ed_8_coef', 0.000664691803386926)
('Ed_9_coef', 0.00028892960680369984)
('Ed_10_coef', 0.00045022530256069314)
('det_entropy_shannon_1_coef', 0.0004597221436949669)
('det_entropy_shannon_2_coef', 0.00034459669021247216)
('det_entropy_shannon_3_coef', 0.0014472478355347562)
('det_entropy_shannon_4_coef', 0.0006513855360284571)
('det_entropy_shannon_5_coef', 0.0009695188485614741)
('det_entropy_shannon_6_coef', 0.0006220564334134669)
('det_entropy_shannon_7_coef', 0.00015671471859099302)
('det_entropy_shannon_8_coef', 0.0005434916930171131)
('det_entropy_shannon_9_coef', 0.0006438571984945853)
('det_entropy_shannon_10_coef', 0.00037126104894809244)
('det_entropy_log_1_coef', 0.0013002686383606678)
('det_entropy_log_2_coef', 0.0010964986340740864)
('det_entropy_log_3_coef', 0.001069331596510022)
('det_entropy_log_4_coef', 0.0012135172545762096)
('det_entropy_log_5_coef', 0.0009130984240153611)
('det_entropy_log_6_coef', 0.0008788116445255772)
('det_entropy_log_7_coef', 0.0004355538281047897)
('det_entropy_log_8_coef', 0.0007874778378806946)
('det_entropy_log_9_coef', 0.0002772771768078367)
('det_entropy_log_10_coef', 0.0003864098498253637)
('det_TKEO_mean_1_coef', 0.0021953675688717775)
('det_TKEO_mean_2_coef', 0.0010519678588933265)
('det_TKEO_mean_3_coef', 0.000825392936714734)
('det_TKEO_mean_4_coef', 0.000386332188125319)
('det_TKEO_mean_5_coef', 0.0005188454975107002)
('det_TKEO_mean_6_coef', 0.0011528283094252578)
('det_TKEO_mean_7_coef', 0.00028793265172548767)
('det_TKEO_mean_8_coef', 0.0007681479569953817)
('det_TKEO_mean_9_coef', 0.00020893407746671653)
('det_TKEO_mean_10_coef', 0.0004344264953808658)
('det_TKEO_std_1_coef', 0.0004933521887269661)
('det_TKEO_std_2_coef', 0.00039879133394430804)
('det_TKEO_std_3_coef', 0.0006330730693760658)
('det_TKEO_std_4_coef', 0.00010188921186773149)
('det_TKEO_std_5_coef', 0.0008295768294623858)
('det_TKEO_std_6_coef', 0.0003131358093384258)
('det_TKEO_std_7_coef', 0.0006917505183396618)
('det_TKEO_std_8_coef', 0.0001397029411833328)
('det_TKEO_std_9_coef', 0.00039298917150148543)
('det_TKEO_std_10_coef', 0.00023260396595915732)
('app_entropy_shannon_1_coef', 0.0003689151340480402)
('app_entropy_shannon_2_coef', 0.0011190751745584248)
('app_entropy_shannon_3_coef', 0.0014519750705693963)
('app_entropy_shannon_4_coef', 0.0005858620055347839)
('app_entropy_shannon_5_coef', 0.00071868978503169)
('app_entropy_shannon_6_coef', 0.000565078498563431)
('app_entropy_shannon_7_coef', 0.000520991420680671)
('app_entropy_shannon_8_coef', 0.0003889771234179068 5)
('app_entropy_shannon_9_coef', 0.0006816331867012953)
('app_entropy_shannon_10_coef', 0.000678268652816511)
('app_entropy_log_1_coef', 0.0014162590957855793)
('app_entropy_log_2_coef', 0.0005606681295179891)
('app_entropy_log_3_coef', 0.0004956065677599882)
('app_entropy_log_4_coef', 0.0006823298843127064)
('app_entropy_log_5_coef', 0.00030583671270672353)
```

```
('app_entropy_log_6_coef', 0.0012689695737339403)
('app_entropy_log_7_coef', 0.002541884232246158)
('app_entropy_log_8_coef', 0.00025019148706671344)
('app_entropy_log_9_coef', 0.0007483506788746792)
('app_entropy_log_10_coef', 0.0003128240955008114)
('app_det_TKEO_mean_1_coef', 0.0008907715401926161)
('app_det_TKEO_mean_2_coef', 0.0020453709663865553)
('app_det_TKEO_mean_3_coef', 0.0009016691203865807)
('app_det_TKEO_mean_4_coef', 0.0018666894533684848)
('app_det_TKEO_mean_5_coef', 0.00026458530840832006)
('app_det_TKEO_mean_6_coef', 0.000531456520201455)
('app_det_TKEO_mean_7_coef', 0.0008685412384239632)
('app_det_TKEO_mean_8_coef', 0.0024352916754735465)
('app_det_TKEO_mean_9_coef', 0.0008292988472140581)
('app_det_TKEO_mean_10_coef', 0.0012396596762433555)
('app_TKEO_std_1_coef', 0.000492463201413606)
('app_TKEO_std_2_coef', 0.001412048393907995)
('app_TKEO_std_3_coef', 0.0009058399712719786)
('app_TKEO_std_4_coef', 0.0007351589345959959)
('app_TKEO_std_5_coef', 0.001110341385451375)
('app_TKEO_std_6_coef', 0.0008641023470628581)
('app_TKEO_std_7_coef', 0.001672564919952019)
('app_TKEO_std_8_coef', 0.0010194521524625104)
('app_TKEO_std_9_coef', 0.0004171652471382661)
('app_TKEO_std_10_coef', 0.0004988981490419199)
('Ea2', 0.0003018016932482482)
('Ed2_1_coef', 0.0012867386159686174)
('Ed2_2_coef', 0.00034729625950878016)
('Ed2_3_coef', 0.00101526820992725)
('Ed2_4_coef', 0.0015172525223989164)
('Ed2_5_coef', 0.0011845257386414929)
('Ed2_6_coef', 0.0004482698701016383)
('Ed2_7_coef', 0.0012968886927670544)
('Ed2_8_coef', 0.0007831436975661841)
('Ed2_9_coef', 0.0002558983834018288)
('Ed2_10_coef', 0.00045930889282870866)
('det_LT_entropy_shannon_1_coef', 0.0001889247614814486)
('det_LT_entropy_shannon_2_coef', 0.0007361135074663589)
('det_LT_entropy_shannon_3_coef', 0.0006035479142982203)
('det_LT_entropy_shannon_4_coef', 0.0006122600393417655)
('det_LT_entropy_shannon_5_coef', 0.0011516312195750675)
('det_LT_entropy_shannon_6_coef', 0.0006695302781317659)
('det_LT_entropy_shannon_7_coef', 0.0007696292936543884)
('det_LT_entropy_shannon_8_coef', 0.0006235099773687294)
('det_LT_entropy_shannon_9_coef', 0.00018895359705173656)
('det_LT_entropy_shannon_10_coef', 0.00030803818775040323)
('det_LT_entropy_log_1_coef', 0.001225470272209118)
('det_LT_entropy_log_2_coef', 0.0007458314833880168)
('det_LT_entropy_log_3_coef', 0.000640027187539093)
('det_LT_entropy_log_4_coef', 0.00042522603837223276)
('det_LT_entropy_log_5_coef', 0.0006746949132849616)
('det_LT_entropy_log_6_coef', 0.0006071290202415266)
('det_LT_entropy_log_7_coef', 0.0009042902309772658)
('det_LT_entropy_log_8_coef', 0.0009125867435909364)
('det_LT_entropy_log_9_coef', 0.00018195012621213244)
('det_LT_entropy_log_10_coef', 0.00016903226938665838)
('det_LT_TKEO_mean_1_coef', 0.0005934186375430265)
('det_LT_TKEO_mean_2_coef', 0.0006273863854778037)
('det_LT_TKEO_mean_3_coef', 0.00039434901055209864)
('det_LT_TKEO_mean_4_coef', 0.0006033725112839222)
('det_LT_TKEO_mean_5_coef', 0.0008077418999778804)
('det_LT_TKEO_mean_6_coef', 0.00052328018739903 1)
('det_LT_TKEO_mean_7_coef', 0.0006644145918189478)
('det_LT_TKEO_mean_8_coef', 0.0002879445377587243)
('det_LT_TKEO_mean_9_coef', 0.0006020227578693452)
('det_LT_TKEO_mean_10_coef', 0.0005101382539419259)
('det_LT_TKEO_std_1_coef', 0.0008671344999574705)
('det_LT_TKEO_std_2_coef', 0.0003074250788402192)
('det_LT_TKEO_std_3_coef', 0.0004761526737177104)
```

```
('det_LT_TKEO_std_4_coef', 0.0002248476641092981)
('det_LT_TKEO_std_5_coef', 0.0009015421750729667)
('det_LT_TKEO_std_6_coef', 0.0013715702524731163)
('det_LT_TKEO_std_7_coef', 0.0003779017038993141)
('det_LT_TKEO_std_8_coef', 0.0007860952734608602)
('det_LT_TKEO_std_9_coef', 0.000457629540538751)
('det_LT_TKEO_std_10_coef', 0.0004803551245519085)
('app_LT_entropy_shannon_1_coef', 0.00119811585527443)
('app_LT_entropy_shannon_2_coef', 0.0008629790206654902)
('app_LT_entropy_shannon_3_coef', 0.000380836280601488)
('app_LT_entropy_shannon_4_coef', 0.0004907658327754012)
('app_LT_entropy_shannon_5_coef', 0.0009645909516022464)
('app_LT_entropy_shannon_6_coef', 0.0003996166529204574)
('app_LT_entropy_shannon_7_coef', 0.0002076991823539538)
('app_LT_entropy_shannon_8_coef', 0.000939212526264776)
('app_LT_entropy_shannon_9_coef', 0.0005845368136101878)
('app_LT_entropy_shannon_10_coef', 0.0008408412262307189)
('app_LT_entropy_log_1_coef', 0.0015735941734594172)
('app_LT_entropy_log_2_coef', 0.00023320068903923773)
('app_LT_entropy_log_3_coef', 0.00048608095194664945)
('app_LT_entropy_log_4_coef', 0.0005280986731472768)
('app_LT_entropy_log_5_coef', 0.0005327729386058983)
('app_LT_entropy_log_6_coef', 0.0008413224620323798)
('app_LT_entropy_log_7_coef', 8.798997244179178e-05)
('app_LT_entropy_log_8_coef', 0.0007282782544122914)
('app_LT_entropy_log_9_coef', 0.001893366821869793)
('app_LT_entropy_log_10_coef', 0.0009417393037236233)
('app_LT_TKEO_mean_1_coef', 0.0006708818337418164)
('app_LT_TKEO_mean_2_coef', 0.0005863023880465657)
('app_LT_TKEO_mean_3_coef', 0.0005862359622360563)
('app_LT_TKEO_mean_4_coef', 0.0003479237409454838)
('app_LT_TKEO_mean_5_coef', 0.001302010086619853)
('app_LT_TKEO_mean_6_coef', 0.0009821493028284015)
('app_LT_TKEO_mean_7_coef', 0.001483279980866948)
('app_LT_TKEO_mean_8_coef', 0.0005342400850624811)
('app_LT_TKEO_mean_9_coef', 0.0007681404893365871)
('app_LT_TKEO_mean_10_coef', 0.001348072047870179)
('app_LT_TKEO_std_1_coef', 0.0010209902362889618)
('app_LT_TKEO_std_2_coef', 0.001062700760693611)
('app_LT_TKEO_std_3_coef', 0.0009959837393962664)
('app_LT_TKEO_std_4_coef', 0.0005413131474150336)
('app_LT_TKEO_std_5_coef', 0.0004402950953945856)
('app_LT_TKEO_std_6_coef', 0.0013294715310849397)
('app_LT_TKEO_std_7_coef', 0.00039041883059522553)
('app_LT_TKEO_std_8_coef', 0.00019231961168723403)
('app_LT_TKEO_std_9_coef', 0.0017772375377939405)
('app_LT_TKEO_std_10_coef', 0.000821994950746797)
('tqwt_energy_dec_1', 0.000825572847010409)
('tqwt_energy_dec_2', 0.0012794817810822727)
('tqwt_energy_dec_3', 0.0011590771587259687)
('tqwt_energy_dec_4', 0.0014901524127417308)
('tqwt_energy_dec_5', 0.0007486805894889964)
('tqwt_energy_dec_6', 0.0024229241463260423)
('tqwt_energy_dec_7', 0.0007654266272907191)
('tqwt_energy_dec_8', 0.0005368972444354795)
('tqwt_energy_dec_9', 0.0003294925018174935)
('tqwt_energy_dec_10', 0.0020331926065289543)
('tqwt_energy_dec_11', 0.00939823379432209)
('tqwt_energy_dec_12', 0.0030534508650554766)
('tqwt_energy_dec_13', 0.0012549773436266317)
('tqwt_energy_dec_14', 0.0010352406312748265)
('tqwt_energy_dec_15', 0.004327029217892845)
('tqwt_energy_dec_16', 0.003194593263215912)
('tqwt_energy_dec_17', 0.0017721043405910112)
('tqwt_energy_dec_18', 0.0031085327081763815)
('tqwt_energy_dec_19', 0.0009737553290627225)
('tqwt_energy_dec_20', 0.0002852609125848954)
('tqwt_energy_dec_21', 0.0017902579348561783)
('tqwt_energy_dec_22', 0.0009885981611459382)
```

```
('tqwt_energy_dec_23', 0.0009159826078361506)
('tqwt_energy_dec_24', 0.0012174858523559284)
('tqwt_energy_dec_25', 0.002750130292822633)
('tqwt_energy_dec_26', 0.00584874472192575)
('tqwt_energy_dec_27', 0.005435990345780379)
('tqwt_energy_dec_28', 0.002805868937141667)
('tqwt_energy_dec_29', 0.0007456866225782983)
('tqwt_energy_dec_30', 0.0006118570958203811)
('tqwt_energy_dec_31', 0.00042081253009178254)
('tqwt_energy_dec_32', 0.0009540600282128024)
('tqwt_energy_dec_33', 0.001689416785739209)
('tqwt_energy_dec_34', 0.0006610283470443426)
('tqwt_energy_dec_35', 0.002248948446179085)
('tqwt_energy_dec_36', 0.0005949833390306382)
('tqwt_entropy_shannon_dec_1', 0.001994740373907156)
('tqwt_entropy_shannon_dec_2', 0.0006432348477851627)
('tqwt_entropy_shannon_dec_3', 0.0008065130584196946)
('tqwt_entropy_shannon_dec_4', 0.001148593094941047)
('tqwt_entropy_shannon_dec_5', 0.0014427704468909439)
('tqwt_entropy_shannon_dec_6', 0.001625959638910099)
('tqwt_entropy_shannon_dec_7', 0.0010544988037864934)
('tqwt_entropy_shannon_dec_8', 0.0013410332712184582)
('tqwt_entropy_shannon_dec_9', 0.0023247667159044484)
('tqwt_entropy_shannon_dec_10', 0.0023672298438095147)
('tqwt_entropy_shannon_dec_11', 0.004846420119935647)
('tqwt_entropy_shannon_dec_12', 0.013462942683508516)
('tqwt_entropy_shannon_dec_13', 0.005062522079800612)
('tqwt_entropy_shannon_dec_14', 0.002116055864081791)
('tqwt_entropy_shannon_dec_15', 0.001523989249781099)
('tqwt_entropy_shannon_dec_16', 0.0012986757159361626)
('tqwt_entropy_shannon_dec_17', 0.002051054126382253)
('tqwt_entropy_shannon_dec_18', 0.0007860169950441628)
('tqwt_entropy_shannon_dec_19', 0.002165754575242521)
('tqwt_entropy_shannon_dec_20', 0.000758199060162793)
('tqwt_entropy_shannon_dec_21', 0.0010379381710622978)
('tqwt_entropy_shannon_dec_22', 0.00035429764357019074)
('tqwt_entropy_shannon_dec_23', 0.0004612418573693943)
('tqwt_entropy_shannon_dec_24', 0.0009423765973398172)
('tqwt_entropy_shannon_dec_25', 0.000953913812627355)
('tqwt_entropy_shannon_dec_26', 0.001190265764356743)
('tqwt_entropy_shannon_dec_27', 0.0012087084087760013)
('tqwt_entropy_shannon_dec_28', 0.0009486822740842937)
('tqwt_entropy_shannon_dec_29', 0.0007825908620526675)
('tqwt_entropy_shannon_dec_30', 0.0009805639223523792)
('tqwt_entropy_shannon_dec_31', 0.0010098743427495293)
('tqwt_entropy_shannon_dec_32', 0.0011863878609854608)
('tqwt_entropy_shannon_dec_33', 0.002947680080758208)
('tqwt_entropy_shannon_dec_34', 0.004259288690870935)
('tqwt_entropy_shannon_dec_35', 0.0016201843192936163)
('tqwt_entropy_shannon_dec_36', 0.0027220461637089055)
('tqwt_entropy_log_dec_1', 0.0022366260393657327)
('tqwt_entropy_log_dec_2', 0.0006495464066416921)
('tqwt_entropy_log_dec_3', 0.00081336571989038)
('tqwt_entropy_log_dec_4', 0.0007043743364011888)
('tqwt_entropy_log_dec_5', 0.0011642910839915986)
('tqwt_entropy_log_dec_6', 0.000563492830967646)
('tqwt_entropy_log_dec_7', 0.000724795765212931)
('tqwt_entropy_log_dec_8', 0.0012507841024924507)
('tqwt_entropy_log_dec_9', 0.0006284991045356859)
('tqwt_entropy_log_dec_10', 0.0009048974256974396)
('tqwt_entropy_log_dec_11', 0.005262984545293315)
('tqwt_entropy_log_dec_12', 0.007916144932504364)
('tqwt_entropy_log_dec_13', 0.0031078766708881036)
('tqwt_entropy_log_dec_14', 0.001286022806153528)
('tqwt_entropy_log_dec_15', 0.0006465284577642058)
('tqwt_entropy_log_dec_16', 0.001985772923335067)
('tqwt_entropy_log_dec_17', 0.0021413559689066847)
('tqwt_entropy_log_dec_18', 0.0009144409468648901)
('tqwt_entropy_log_dec_19', 0.0030669523822112865)
```

```
('tqwt_entropy_log_dec_20', 0.0007822473302030971)
('tqwt_entropy_log_dec_21', 0.0006967263073740822)
('tqwt_entropy_log_dec_22', 0.0008137633377311746)
('tqwt_entropy_log_dec_23', 0.0008228260199280576)
('tqwt_entropy_log_dec_24', 0.0013356150760224477)
('tqwt_entropy_log_dec_25', 0.0018553524630871834)
('tqwt_entropy_log_dec_26', 0.002197524929769903)
('tqwt_entropy_log_dec_27', 0.00510735849977353)
('tqwt_entropy_log_dec_28', 0.0019652586899627233)
('tqwt_entropy_log_dec_29', 0.0019081751213989645)
('tqwt_entropy_log_dec_30', 0.0005779125264509418)
('tqwt_entropy_log_dec_31', 0.0013235054190840524)
('tqwt_entropy_log_dec_32', 0.0005166256772664735)
('tqwt_entropy_log_dec_33', 0.0020170342037169778)
('tqwt_entropy_log_dec_34', 0.0011380770280626064)
('tqwt_entropy_log_dec_35', 0.004511936105836038)
('tqwt_entropy_log_dec_36', 0.0008096261469773393)
('tqwt_TKEO_mean_dec_1', 0.0011191048141516058)
('tqwt_TKEO_mean_dec_2', 0.00041304174956282875)
('tqwt_TKEO_mean_dec_3', 0.0010111328593638196)
('tqwt_TKEO_mean_dec_4', 0.0008401145301887556)
('tqwt_TKEO_mean_dec_5', 0.0013443277316507135)
('tqwt_TKEO_mean_dec_6', 0.0025437219169992067)
('tqwt_TKEO_mean_dec_7', 0.0025310465241960727)
('tqwt_TKEO_mean_dec_8', 0.0006538967965764315)
('tqwt_TKEO_mean_dec_9', 0.000646712249857176)
('tqwt_TKEO_mean_dec_10', 0.0014247646097513 63)
('tqwt_TKEO_mean_dec_11', 0.004493228839951114)
('tqwt_TKEO_mean_dec_12', 0.012386995402238227)
('tqwt_TKEO_mean_dec_13', 0.002858448577942048)
('tqwt_TKEO_mean_dec_14', 0.003897786262516919)
('tqwt_TKEO_mean_dec_15', 0.00035613652060712035)
('tqwt_TKEO_mean_dec_16', 0.0021631072921593263)
('tqwt_TKEO_mean_dec_17', 0.002175847649131404)
('tqwt_TKEO_mean_dec_18', 0.0014915570851712638)
('tqwt_TKEO_mean_dec_19', 0.002294775012341647)
('tqwt_TKEO_mean_dec_20', 0.000479509195047218)
('tqwt_TKEO_mean_dec_21', 0.00044515834851321646)
('tqwt_TKEO_mean_dec_22', 0.0003937596093801638)
('tqwt_TKEO_mean_dec_23', 0.0009628175480503892)
('tqwt_TKEO_mean_dec_24', 0.00019700942676662326)
('tqwt_TKEO_mean_dec_25', 0.0009997587313369867)
('tqwt_TKEO_mean_dec_26', 0.0018910485160916098)
('tqwt_TKEO_mean_dec_27', 0.0016328056759868453)
('tqwt_TKEO_mean_dec_28', 0.0013080422963436932)
('tqwt_TKEO_mean_dec_29', 0.000574682782833287)
('tqwt_TKEO_mean_dec_30', 0.0005638312816240956)
('tqwt_TKEO_mean_dec_31', 0.0007471633497636056)
('tqwt_TKEO_mean_dec_32', 0.0012054605043389191)
('tqwt_TKEO_mean_dec_33', 0.0032487219922156694)
('tqwt_TKEO_mean_dec_34', 0.0017499995320458463)
('tqwt_TKEO_mean_dec_35', 0.0012337885650225882)
('tqwt_TKEO_mean_dec_36', 0.0018285559384125882)
('tqwt_TKEO_std_dec_1', 0.001229699864939815)
('tqwt_TKEO_std_dec_2', 0.0013611885962414565)
('tqwt_TKEO_std_dec_3', 0.0008605707142902219)
('tqwt_TKEO_std_dec_4', 0.00028046805914763186)
('tqwt_TKEO_std_dec_5', 0.0022056395406820574)
('tqwt_TKEO_std_dec_6', 0.0026724428343016427)
('tqwt_TKEO_std_dec_7', 0.0009815476647645472)
('tqwt_TKEO_std_dec_8', 0.00031722198404501113)
('tqwt_TKEO_std_dec_9', 0.0019768248735733433)
('tqwt_TKEO_std_dec_10', 0.000912540909881689)
('tqwt_TKEO_std_dec_11', 0.006564293805751928)
('tqwt_TKEO_std_dec_12', 0.009075829155213258)
('tqwt_TKEO_std_dec_13', 0.0057529085379626095)
('tqwt_TKEO_std_dec_14', 0.0016946714248543802)
('tqwt_TKEO_std_dec_15', 0.0005783951840165278)
('tqwt_TKEO_std_dec_16', 0.0029376269596409813)
```

```
('tqwt_TKEO_std_dec_17', 0.0021192815147051875)
('tqwt_TKEO_std_dec_18', 0.001892675489855391)
('tqwt_TKEO_std_dec_19', 0.0027235173590664757)
('tqwt_TKEO_std_dec_20', 0.002069546397208942)
('tqwt_TKEO_std_dec_21', 0.001044962290682804)
('tqwt_TKEO_std_dec_22', 0.0010926970378574796)
('tqwt_TKEO_std_dec_23', 0.0005693628617373701)
('tqwt_TKEO_std_dec_24', 0.0009960358213157168)
('tqwt_TKEO_std_dec_25', 0.002508881320747152)
('tqwt_TKEO_std_dec_26', 0.0009626539178491997)
('tqwt_TKEO_std_dec_27', 0.0011473973540882091)
('tqwt_TKEO_std_dec_28', 0.0007361745893329799)
('tqwt_TKEO_std_dec_29', 0.00036721473945893195)
('tqwt_TKEO_std_dec_30', 0.0005812948247309906)
('tqwt_TKEO_std_dec_31', 0.0005276574586190733)
('tqwt_TKEO_std_dec_32', 0.0007310705825433087)
('tqwt_TKEO_std_dec_33', 0.0010576948500877089)
('tqwt_TKEO_std_dec_34', 0.0009034776855796312)
('tqwt_TKEO_std_dec_35', 0.0007653749245947576)
('tqwt_TKEO_std_dec_36', 0.000493193113799378)
('tqwt_medianValue_dec_1', 0.0008081620016374866)
('tqwt_medianValue_dec_2', 0.0006504405392425076)
('tqwt_medianValue_dec_3', 0.0011160246871277843)
('tqwt_medianValue_dec_4', 0.0005325273661451085)
('tqwt_medianValue_dec_5', 0.0002488713771038166)
('tqwt_medianValue_dec_6', 0.0007842329681798573)
('tqwt_medianValue_dec_7', 0.0006873575294784733)
('tqwt_medianValue_dec_8', 0.0014314564047467203)
('tqwt_medianValue_dec_9', 0.0012654518532012882)
('tqwt_medianValue_dec_10', 0.0004533000697418611)
('tqwt_medianValue_dec_11', 0.000886755306869053)
('tqwt_medianValue_dec_12', 0.0009296199856696623)
('tqwt_medianValue_dec_13', 0.0013437638274430938)
('tqwt_medianValue_dec_14', 0.0008989718231329981)
('tqwt_medianValue_dec_15', 0.0005738575099394964)
('tqwt_medianValue_dec_16', 0.0013429031787016384)
('tqwt_medianValue_dec_17', 0.0007745424217282212)
('tqwt_medianValue_dec_18', 0.0010460962809914862)
('tqwt_medianValue_dec_19', 0.0012273320578747393)
('tqwt_medianValue_dec_20', 0.0010136662686301282)
('tqwt_medianValue_dec_21', 0.0015844672566289492)
('tqwt_medianValue_dec_22', 0.0014845202005744813)
('tqwt_medianValue_dec_23', 0.0013039971480583955)
('tqwt_medianValue_dec_24', 0.0007257223868888554)
('tqwt_medianValue_dec_25', 0.0012408439828228226)
('tqwt_medianValue_dec_26', 0.0008180442389694708)
('tqwt_medianValue_dec_27', 0.00036545017034059224)
('tqwt_medianValue_dec_28', 0.0007488960889731385)
('tqwt_medianValue_dec_29', 0.0009851379142654531)
('tqwt_medianValue_dec_30', 0.00014858222736125587)
('tqwt_medianValue_dec_31', 0.0018722620981216666)
('tqwt_medianValue_dec_32', 0.0010766566691648193)
('tqwt_medianValue_dec_33', 0.0013704900571128136)
('tqwt_medianValue_dec_34', 0.00048063622510251643)
('tqwt_medianValue_dec_35', 0.00028019720202696334)
('tqwt_medianValue_dec_36', 0.0016157201408526474)
('tqwt_meanValue_dec_1', 0.00027961565357376535)
('tqwt_meanValue_dec_2', 0.0003941382131296038 5)
('tqwt_meanValue_dec_3', 0.0006042750226323186)
('tqwt_meanValue_dec_4', 0.0003583261895834223)
('tqwt_meanValue_dec_5', 0.0012365046673805879)
('tqwt_meanValue_dec_6', 0.0012570791569759257)
('tqwt_meanValue_dec_7', 0.0007648308356286326)
('tqwt_meanValue_dec_8', 0.00035433604385834324)
('tqwt_meanValue_dec_9', 0.0007460726480004585)
('tqwt_meanValue_dec_10', 0.0005753005777578651)
('tqwt_meanValue_dec_11', 0.0013147108359132212)
('tqwt_meanValue_dec_12', 0.0012615751572328743)
('tqwt_meanValue_dec_13', 0.0008020389100144632)
```

```
('tqwt_meanValue_dec_14', 0.00024265091510088788)
('tqwt_meanValue_dec_15', 0.000669952241360585)
('tqwt_meanValue_dec_16', 0.00108370721038401 7)
('tqwt_meanValue_dec_17', 0.00164318515587207 6)
('tqwt_meanValue_dec_18', 0.00171208506158835 37)
('tqwt_meanValue_dec_19', 0.00052979514186173 12)
('tqwt_meanValue_dec_20', 0.00108581216193220 55)
('tqwt_meanValue_dec_21', 0.000651205521011924)
('tqwt_meanValue_dec_22', 0.00128312242218977 4)
('tqwt_meanValue_dec_23', 0.00089705218474183 33)
('tqwt_meanValue_dec_24', 0.00077707642015188 55)
('tqwt_meanValue_dec_25', 0.00079634925944191 46)
('tqwt_meanValue_dec_26', 0.00046158025127283556)
('tqwt_meanValue_dec_27', 0.001101768566285425)
('tqwt_meanValue_dec_28', 0.00027518510256449 38)
('tqwt_meanValue_dec_29', 0.00063128115503500 39)
('tqwt_meanValue_dec_30', 0.00018489450455119426)
('tqwt_meanValue_dec_31', 0.00032070823925479605)
('tqwt_meanValue_dec_32', 0.0007992752158405807)
('tqwt_meanValue_dec_33', 0.002315248801268484)
('tqwt_meanValue_dec_34', 0.0017744701158534204)
('tqwt_meanValue_dec_35', 0.0005392573315546493)
('tqwt_meanValue_dec_36', 0.003452526102547696)
('tqwt_stdValue_dec_1', 0.001741488772568137)
('tqwt_stdValue_dec_2', 0.0004997430454821975)
('tqwt_stdValue_dec_3', 0.0011278496500919184)
('tqwt_stdValue_dec_4', 0.0015649206575786584)
('tqwt_stdValue_dec_5', 0.0003391820130371707)
('tqwt_stdValue_dec_6', 0.002770944723677935)
('tqwt_stdValue_dec_7', 0.0023846713726627727)
('tqwt_stdValue_dec_8', 0.0010459743176269869)
('tqwt_stdValue_dec_9', 0.0008424508950101404)
('tqwt_stdValue_dec_10', 0.001217076002038602)
('tqwt_stdValue_dec_11', 0.008833428904930938)
('tqwt_stdValue_dec_12', 0.005997522212791165)
('tqwt_stdValue_dec_13', 0.0056391887053846785)
('tqwt_stdValue_dec_14', 0.002485464840981794)
('tqwt_stdValue_dec_15', 0.0022846897520043233)
('tqwt_stdValue_dec_16', 0.001339830131415515)
('tqwt_stdValue_dec_17', 0.001704342192081889)
('tqwt_stdValue_dec_18', 0.002109346081600738)
('tqwt_stdValue_dec_19', 0.003219196760727127)
('tqwt_stdValue_dec_20', 0.0006007725223054619)
('tqwt_stdValue_dec_21', 0.0009156580753899284)
('tqwt_stdValue_dec_22', 0.0005476790035507451)
('tqwt_stdValue_dec_23', 0.0010431206025178377)
('tqwt_stdValue_dec_24', 0.000560895455750284)
('tqwt_stdValue_dec_25', 0.0009927158789703606)
('tqwt_stdValue_dec_26', 0.0010730723971022916)
('tqwt_stdValue_dec_27', 0.001246296286146234)
('tqwt_stdValue_dec_28', 0.001039608668196527)
('tqwt_stdValue_dec_29', 0.0004324985065377634)
('tqwt_stdValue_dec_30', 0.0007855343140685182)
('tqwt_stdValue_dec_31', 0.000648121444325038)
('tqwt_stdValue_dec_32', 0.0011478524360659693)
('tqwt_stdValue_dec_33', 0.0016465497948027481)
('tqwt_stdValue_dec_34', 0.0014095239265543831)
('tqwt_stdValue_dec_35', 0.0015553173638039848)
('tqwt_stdValue_dec_36', 0.0025130674676037772)
('tqwt_minValue_dec_1', 0.0004831501279069424)
('tqwt_minValue_dec_2', 0.0008733230686101869)
('tqwt_minValue_dec_3', 0.0002370581312237963)
('tqwt_minValue_dec_4', 0.0009032113447557411)
('tqwt_minValue_dec_5', 0.0014762664643577497)
('tqwt_minValue_dec_6', 0.0011512702393774185)
('tqwt_minValue_dec_7', 0.0009118000269940365)
('tqwt_minValue_dec_8', 0.000788152097255373)
('tqwt_minValue_dec_9', 0.0012291864360841444)
('tqwt_minValue_dec_10', 0.0013461911796074921)
```

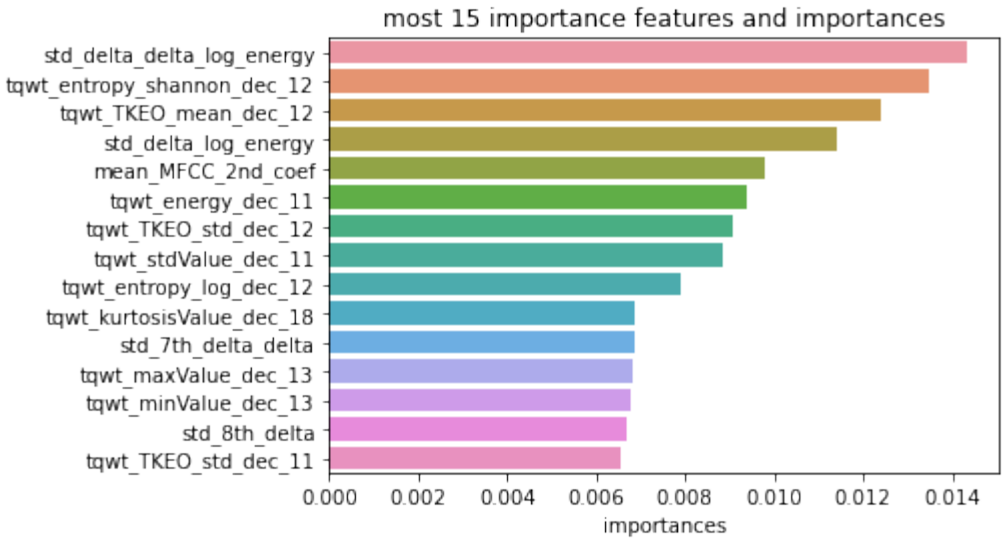
```
('tqwt_minValue_dec_11', 0.003965861896178216)
('tqwt_minValue_dec_12', 0.003492535742860506)
('tqwt_minValue_dec_13', 0.006760891991791312)
('tqwt_minValue_dec_14', 0.003168101162218519)
('tqwt_minValue_dec_15', 0.0008249337591072376)
('tqwt_minValue_dec_16', 0.0004918298650449567)
('tqwt_minValue_dec_17', 0.002730536580962029)
('tqwt_minValue_dec_18', 0.00134888897849576)
('tqwt_minValue_dec_19', 0.00223324334369065)
('tqwt_minValue_dec_20', 0.00029590444617198303)
('tqwt_minValue_dec_21', 0.0010778857807911763)
('tqwt_minValue_dec_22', 0.0005330691971589549)
('tqwt_minValue_dec_23', 0.0007413405645054161)
('tqwt_minValue_dec_24', 0.0009316319091466714)
('tqwt_minValue_dec_25', 0.0008028699521273328)
('tqwt_minValue_dec_26', 0.0015723228851749864)
('tqwt_minValue_dec_27', 0.0006468994875453686)
('tqwt_minValue_dec_28', 0.00032105216573960076)
('tqwt_minValue_dec_29', 0.00029618449773234103)
('tqwt_minValue_dec_30', 0.000332228100906884)
('tqwt_minValue_dec_31', 0.0001817702358346698)
('tqwt_minValue_dec_32', 8.74983444021873e-05)
('tqwt_minValue_dec_33', 0.000988035581657234)
('tqwt_minValue_dec_34', 0.0012549184822609897)
('tqwt_minValue_dec_35', 0.0004592519370311995)
('tqwt_minValue_dec_36', 0.0014292439990808245)
('tqwt_maxValue_dec_1', 0.00042114031727135846)
('tqwt_maxValue_dec_2', 0.0005290147762022232)
('tqwt_maxValue_dec_3', 0.0020491156470369937)
('tqwt_maxValue_dec_4', 0.0017572483812695963)
('tqwt_maxValue_dec_5', 0.0014737282428700196)
('tqwt_maxValue_dec_6', 0.002027601275449324)
('tqwt_maxValue_dec_7', 0.001227792781650675)
('tqwt_maxValue_dec_8', 0.00105412800048951)
('tqwt_maxValue_dec_9', 0.0008109111561170761)
('tqwt_maxValue_dec_10', 0.0004436503208357794)
('tqwt_maxValue_dec_11', 0.0023953615679966436)
('tqwt_maxValue_dec_12', 0.0037273906644325104)
('tqwt_maxValue_dec_13', 0.006812013427300944)
('tqwt_maxValue_dec_14', 0.0007178053215953391)
('tqwt_maxValue_dec_15', 0.001806164206934005)
('tqwt_maxValue_dec_16', 0.0009428678562108587)
('tqwt_maxValue_dec_17', 0.00127638862584786)
('tqwt_maxValue_dec_18', 0.0017474977095181317)
('tqwt_maxValue_dec_19', 0.0013165781978755388)
('tqwt_maxValue_dec_20', 0.0006719051172995869)
('tqwt_maxValue_dec_21', 0.0006828205548814272)
('tqwt_maxValue_dec_22', 0.0003748613472844693)
('tqwt_maxValue_dec_23', 0.0002354766392617102)
('tqwt_maxValue_dec_24', 0.0004302237506722545 3)
('tqwt_maxValue_dec_25', 0.0015811872056162252)
('tqwt_maxValue_dec_26', 0.0016609907416502436)
('tqwt_maxValue_dec_27', 0.0014557153102535642)
('tqwt_maxValue_dec_28', 0.0008354595717258435)
('tqwt_maxValue_dec_29', 0.0005071783850850387)
('tqwt_maxValue_dec_30', 3.89506613491737e-05)
('tqwt_maxValue_dec_31', 0.0007049828859824162)
('tqwt_maxValue_dec_32', 0.000573286929040609)
('tqwt_maxValue_dec_33', 0.00092285869088622)
('tqwt_maxValue_dec_34', 0.0011624094149499415)
('tqwt_maxValue_dec_35', 0.0010807482586398058)
('tqwt_maxValue_dec_36', 0.0016642480007996781)
('tqwt_skewnessValue_dec_1', 0.0005272497264337826)
('tqwt_skewnessValue_dec_2', 0.0005416594820737642)
('tqwt_skewnessValue_dec_3', 0.0005533329528030595)
('tqwt_skewnessValue_dec_4', 0.00038896712814404815)
('tqwt_skewnessValue_dec_5', 0.0003591324507564401)
('tqwt_skewnessValue_dec_6', 0.0008243458776122665)
('tqwt_skewnessValue_dec_7', 0.0005236623978275347)
```

```
('tqwt_skewnessValue_dec_8', 0.0005965710999070351)
('tqwt_skewnessValue_dec_9', 0.0014893324811772234)
('tqwt_skewnessValue_dec_10', 0.0012102943362987508)
('tqwt_skewnessValue_dec_11', 0.0008411818113450878)
('tqwt_skewnessValue_dec_12', 0.0017831832509132893)
('tqwt_skewnessValue_dec_13', 0.0008226073662161043)
('tqwt_skewnessValue_dec_14', 0.000245990272611441)
('tqwt_skewnessValue_dec_15', 0.0011041637469960362)
('tqwt_skewnessValue_dec_16', 0.0007356518330792981)
('tqwt_skewnessValue_dec_17', 0.001380250407066857)
('tqwt_skewnessValue_dec_18', 0.0006449549008915049)
('tqwt_skewnessValue_dec_19', 0.00033396235112289306)
('tqwt_skewnessValue_dec_20', 0.0006993257463202902)
('tqwt_skewnessValue_dec_21', 0.0002124769797782603)
('tqwt_skewnessValue_dec_22', 0.00048169893018792497)
('tqwt_skewnessValue_dec_23', 0.0005784862566761377)
('tqwt_skewnessValue_dec_24', 0.0008575699788005366)
('tqwt_skewnessValue_dec_25', 0.0022330850492491327)
('tqwt_skewnessValue_dec_26', 0.0007653844496275521)
('tqwt_skewnessValue_dec_27', 0.0016276216234859344)
('tqwt_skewnessValue_dec_28', 0.001306093434690283)
('tqwt_skewnessValue_dec_29', 0.001465775196974704)
('tqwt_skewnessValue_dec_30', 0.0013075102211662771)
('tqwt_skewnessValue_dec_31', 0.0003619611528112674)
('tqwt_skewnessValue_dec_32', 0.0003229733990601045)
('tqwt_skewnessValue_dec_33', 0.000613624188687352)
('tqwt_skewnessValue_dec_34', 0.0011942079761838988)
('tqwt_skewnessValue_dec_35', 0.0004292579530188991)
('tqwt_skewnessValue_dec_36', 0.0008959840083231846)
('tqwt_kurtosisValue_dec_1', 0.0005044812185820202)
('tqwt_kurtosisValue_dec_2', 0.00047817676396027056)
('tqwt_kurtosisValue_dec_3', 0.00010563987813187907)
('tqwt_kurtosisValue_dec_4', 0.001002658911282609)
('tqwt_kurtosisValue_dec_5', 0.0006574813240072972)
('tqwt_kurtosisValue_dec_6', 0.0008868200007696904)
('tqwt_kurtosisValue_dec_7', 0.0008895221568505883)
('tqwt_kurtosisValue_dec_8', 0.0007901244194563644)
('tqwt_kurtosisValue_dec_9', 0.001200765165680675)
('tqwt_kurtosisValue_dec_10', 0.000766634099714729)
('tqwt_kurtosisValue_dec_11', 0.0007971710709609423)
('tqwt_kurtosisValue_dec_12', 0.0012739151119148821)
('tqwt_kurtosisValue_dec_13', 0.0009676867979506622)
('tqwt_kurtosisValue_dec_14', 0.0006736730977071609)
('tqwt_kurtosisValue_dec_15', 0.0007457228248659181)
('tqwt_kurtosisValue_dec_16', 0.0007105387308110388)
('tqwt_kurtosisValue_dec_17', 0.002451585220317447)
('tqwt_kurtosisValue_dec_18', 0.006887091491691372)
('tqwt_kurtosisValue_dec_19', 0.002506562684967658)
('tqwt_kurtosisValue_dec_20', 0.0020153861867397535)
('tqwt_kurtosisValue_dec_21', 0.001168175593489728)
('tqwt_kurtosisValue_dec_22', 0.0014598587917266255)
('tqwt_kurtosisValue_dec_23', 0.001975955123756548)
('tqwt_kurtosisValue_dec_24', 0.0009151818484780623)
('tqwt_kurtosisValue_dec_25', 0.000823699559225555)
('tqwt_kurtosisValue_dec_26', 0.0014604680306554187)
('tqwt_kurtosisValue_dec_27', 0.0029019005322162648)
('tqwt_kurtosisValue_dec_28', 0.002350278691451486)
('tqwt_kurtosisValue_dec_29', 0.0006295501847002385)
('tqwt_kurtosisValue_dec_30', 0.0007351349625660917)
('tqwt_kurtosisValue_dec_31', 0.00238779332249237)
('tqwt_kurtosisValue_dec_32', 0.0009763032848947175)
('tqwt_kurtosisValue_dec_33', 0.0013475512477556062)
('tqwt_kurtosisValue_dec_34', 0.002067105076420802)
('tqwt_kurtosisValue_dec_35', 0.001113787268448635)
('tqwt_kurtosisValue_dec_36', 0.0016712539977387865)
```

In [53]:

```
# Get the indices of the most 15 important features
sort_i = np.argsort(clf.feature_importances_ * -1)[:15]
```

```
# Plot the most 15 important features and importances
sns.barplot(y=FEATURES[sort_i], x=clf.feature_importances_[sort_i], orient='h')
plt.xlabel('importances')
plt.title('most 15 importance features and importances')
plt.show()
```



4.2 Identify and select most important features

In [54]:

```
# Create a selector object that will use the random forest classifier to identify features
sfm = SelectFromModel(clf)

# Train the selector
sfm.fit(X_train, y_train)
```

Out[54]:

```
SelectFromModel(estimator=RandomForestClassifier(n_estimators=200, n_jobs=-1,
                                                 random state=33))
```

In [55]:

```
print('Threshold of the feature importance: {}'.format(sfm.threshold_))
```

```
Threshold of the feature importance: 0.0013280212483399738
```

In [56]:

```
# Get the index list of the selected features
selected_i = sfm.get_support(indices=True)
```

In [57]:

```
# Print the name of selected features
IMPORTANT_FEATURES = FEATURES[selected_i]
print(IMPORTANT_FEATURES)
```

```
['PPE' 'DFA' 'locPctJitter' 'locAbsJitter' 'rapJitter' 'ddpJitter'
 'locDbShimmer' 'apq11Shimmer' 'minIntensity' 'meanIntensity' 'f1'
 'GQ_std_cycle_open' 'GNE_std' 'GNE_SNR_TKEO' 'GNE_NSR_TKEO' 'GNE_NSR_SEO'
 'mean_MFCC_0th_coef' 'mean_MFCC_2nd_coef' 'mean_MFCC_3rd_coef'
 'mean_MFCC_5th_coef' 'mean_MFCC_6th_coef' 'mean_MFCC_10th_coef'
 'mean_0th_delta' 'mean_5th_delta' 'mean_10th_delta' 'std_Log_energy'
 'std_MFCC_1st_coef' 'std_MFCC_7th_coef' 'std_MFCC_8th_coef'
 'std_delta_log_energy' 'std_2nd_delta' 'std_4th_delta' 'std_6th_delta'
 'std_7th_delta' 'std_8th_delta' 'std_9th_delta' 'std_10th_delta'
 'std_11th_delta' 'std_12th_delta' 'std_delta_delta_log_energy'
 'std_3rd_delta_delta' 'std_5th_delta_delta' 'std_6th_delta_delta'
```

```
'std_7th_delta_delta' 'std_8th_delta_delta' 'std_9th_delta_delta'
'std_10th_delta_delta' 'std_12th_delta_delta' 'Ed_1_coef' 'Ed_4_coef'
'det_entropy_shannon_3_coef' 'det_TKEO_mean_1_coef'
'app_entropy_shannon_3_coef' 'app_entropy_log_1_coef'
'app_entropy_log_7_coef' 'app_det_TKEO_mean_2_coef'
'app_det_TKEO_mean_4_coef' 'app_det_TKEO_mean_8_coef'
'app_TKEO_std_2_coef' 'app_TKEO_std_7_coef' 'Ed2_4_coef'
'det_LT_TKEO_std_6_coef' 'app_LT_entropy_log_1_coef'
'app_LT_entropy_log_9_coef' 'app_LT_TKEO_mean_7_coef'
'app_LT_TKEO_mean_10_coef' 'app_LT_TKEO_std_6_coef'
'app_LT_TKEO_std_9_coef' 'tqwt_energy_dec_4' 'tqwt_energy_dec_6'
'tqwt_energy_dec_10' 'tqwt_energy_dec_11' 'tqwt_energy_dec_12'
'tqwt_energy_dec_15' 'tqwt_energy_dec_16' 'tqwt_energy_dec_17'
'tqwt_energy_dec_18' 'tqwt_energy_dec_21' 'tqwt_energy_dec_25'
'tqwt_energy_dec_26' 'tqwt_energy_dec_27' 'tqwt_energy_dec_28'
'tqwt_energy_dec_33' 'tqwt_energy_dec_35' 'tqwt_entropy_shannon_dec_1'
'tqwt_entropy_shannon_dec_5' 'tqwt_entropy_shannon_dec_6'
'tqwt_entropy_shannon_dec_8' 'tqwt_entropy_shannon_dec_9'
'tqwt_entropy_shannon_dec_10' 'tqwt_entropy_shannon_dec_11'
'tqwt_entropy_shannon_dec_12' 'tqwt_entropy_shannon_dec_13'
'tqwt_entropy_shannon_dec_14' 'tqwt_entropy_shannon_dec_15'
'tqwt_entropy_shannon_dec_17' 'tqwt_entropy_shannon_dec_19'
'tqwt_entropy_shannon_dec_33' 'tqwt_entropy_shannon_dec_34'
'tqwt_entropy_shannon_dec_35' 'tqwt_entropy_shannon_dec_36'
'tqwt_entropy_log_dec_1' 'tqwt_entropy_log_dec_11'
'tqwt_entropy_log_dec_12' 'tqwt_entropy_log_dec_13'
'tqwt_entropy_log_dec_16' 'tqwt_entropy_log_dec_17'
'tqwt_entropy_log_dec_19' 'tqwt_entropy_log_dec_24'
'tqwt_entropy_log_dec_25' 'tqwt_entropy_log_dec_26'
'tqwt_entropy_log_dec_27' 'tqwt_entropy_log_dec_28'
'tqwt_entropy_log_dec_29' 'tqwt_entropy_log_dec_33'
'tqwt_entropy_log_dec_35' 'tqwt_TKEO_mean_dec_5' 'tqwt_TKEO_mean_dec_6'
'tqwt_TKEO_mean_dec_7' 'tqwt_TKEO_mean_dec_10' 'tqwt_TKEO_mean_dec_11'
'tqwt_TKEO_mean_dec_12' 'tqwt_TKEO_mean_dec_13' 'tqwt_TKEO_mean_dec_14'
'tqwt_TKEO_mean_dec_16' 'tqwt_TKEO_mean_dec_17' 'tqwt_TKEO_mean_dec_18'
'tqwt_TKEO_mean_dec_19' 'tqwt_TKEO_mean_dec_26' 'tqwt_TKEO_mean_dec_27'
'tqwt_TKEO_mean_dec_33' 'tqwt_TKEO_mean_dec_34' 'tqwt_TKEO_mean_dec_36'
'tqwt_TKEO_std_dec_2' 'tqwt_TKEO_std_dec_5' 'tqwt_TKEO_std_dec_6'
'tqwt_TKEO_std_dec_9' 'tqwt_TKEO_std_dec_11' 'tqwt_TKEO_std_dec_12'
'tqwt_TKEO_std_dec_13' 'tqwt_TKEO_std_dec_14' 'tqwt_TKEO_std_dec_16'
'tqwt_TKEO_std_dec_17' 'tqwt_TKEO_std_dec_18' 'tqwt_TKEO_std_dec_19'
'tqwt_TKEO_std_dec_20' 'tqwt_TKEO_std_dec_25' 'tqwt_medianValue_dec_8'
'tqwt_medianValue_dec_13' 'tqwt_medianValue_dec_16'
'tqwt_medianValue_dec_21' 'tqwt_medianValue_dec_22'
'tqwt_medianValue_dec_31' 'tqwt_medianValue_dec_33'
'tqwt_medianValue_dec_36' 'tqwt_meanValue_dec_17' 'tqwt_meanValue_dec_18'
'tqwt_meanValue_dec_33' 'tqwt_meanValue_dec_34' 'tqwt_meanValue_dec_36'
'tqwt_stdValue_dec_1' 'tqwt_stdValue_dec_4' 'tqwt_stdValue_dec_6'
'tqwt_stdValue_dec_7' 'tqwt_stdValue_dec_11' 'tqwt_stdValue_dec_12'
'tqwt_stdValue_dec_13' 'tqwt_stdValue_dec_14' 'tqwt_stdValue_dec_15'
'tqwt_stdValue_dec_16' 'tqwt_stdValue_dec_17' 'tqwt_stdValue_dec_18'
'tqwt_stdValue_dec_19' 'tqwt_stdValue_dec_33' 'tqwt_stdValue_dec_34'
'tqwt_stdValue_dec_35' 'tqwt_stdValue_dec_36' 'tqwt_minValue_dec_5'
'tqwt_minValue_dec_10' 'tqwt_minValue_dec_11' 'tqwt_minValue_dec_12'
'tqwt_minValue_dec_13' 'tqwt_minValue_dec_14' 'tqwt_minValue_dec_17'
'tqwt_minValue_dec_18' 'tqwt_minValue_dec_19' 'tqwt_minValue_dec_26'
'tqwt_minValue_dec_36' 'tqwt_maxValue_dec_3' 'tqwt_maxValue_dec_4'
'tqwt_maxValue_dec_5' 'tqwt_maxValue_dec_6' 'tqwt_maxValue_dec_11'
'tqwt_maxValue_dec_12' 'tqwt_maxValue_dec_13' 'tqwt_maxValue_dec_15'
'tqwt_maxValue_dec_18' 'tqwt_maxValue_dec_25' 'tqwt_maxValue_dec_26'
'tqwt_maxValue_dec_27' 'tqwt_maxValue_dec_36' 'tqwt_skewnessValue_dec_9'
'tqwt_skewnessValue_dec_12' 'tqwt_skewnessValue_dec_17'
'tqwt_skewnessValue_dec_25' 'tqwt_skewnessValue_dec_27'
'tqwt_skewnessValue_dec_29' 'tqwt_kurtosisValue_dec_17'
'tqwt_kurtosisValue_dec_18' 'tqwt_kurtosisValue_dec_19'
'tqwt_kurtosisValue_dec_20' 'tqwt_kurtosisValue_dec_22'
'tqwt_kurtosisValue_dec_23' 'tqwt_kurtosisValue_dec_26'
'tqwt_kurtosisValue_dec_27' 'tqwt_kurtosisValue_dec_28'
'tqwt_kurtosisValue_dec_31' 'tqwt_kurtosisValue_dec_33'
```

```
 'tqwt kurtosisValue dec 34' 'tqwt kurtosisValue dec 36']
```

4.3 Create new dataset with the selected features

In [58]:

```
X_important_train = sfm.transform(X_train)
X_important_test = sfm.transform(X_test)
print('The shape of X_train changed from {} to {}'.format(
    X_train.shape, X_important_train.shape))
print('\nThe shape of X_test changed from {} to {}'.format(
    X_test.shape, X_important_test.shape))
```

```
The shape of X_train changed from (604, 753) to (604, 220)

The shape of X test changed from (152, 753) to (152, 220)
```

5. Extra Trees Classifier Tuning using Random Hyperparameter Search

5.1 Ste up Random Hyperparameter Grid

To use RandomizedSearchCV, we first need to create a parameter grid On each iteration, the algorithm will choose a difference combination of the features. There are overall $5 \quad 6 \quad 4 \quad 3 \quad 3 = 1080$ settings. However, the benefit of a random search is that we are not trying every combination, but selecting at random to sample a wide range of values.

In [59]:

```
# Create Parameter Grid
param_grid = {
    'n_estimators': [100, 150, 200, 250,
                     300],  # Number of trees in random forest
    'max_features': ['auto', 'log2', 0.25, 0.5, 0.75,
                     1],  # Number of features to consider at every split
    'max_depth': [40, 70, 100, None],  # Maximum number of levels in tree
    'min_samples_split':
    [2, 5, 10],  # Minimum number of samples required to split a node
    'min_samples_leaf': [1, 2, 4]
}

pprint(param_grid)
```

```
{'max_depth': [40, 70, 100, None],
 'max_features': ['auto', 'log2', 0.25, 0.5, 0.75, 1],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n estimators': [100, 150, 200, 250, 300]}
```

5.2 Random Search Training

Random search of parameter distributions, using 5-fold cross validation, search acrosss 100 different combinations, and use all available cores for sklearn.ensemble.ExtraTreesClassifier

In [60]:

```
%%time
# Create a Random forest classifier object
xtrees_model = ExtraTreesClassifier(random_state=RANDOM_SEED, n_jobs=-1)

# Create a RandomGridSearchCV object
rs_xtrees = RandomizedSearchCV(xtrees_model,
                               param_distributions=param_grid,
                               n_iter=100,
```

```
                                cv=5,
                                verbose=2,
                                n_jobs=-1,
                                random_state=RANDOM_SEED)

rs_xtrees.fit(X_important_train, y_train)
rs_xtrees.best_params_, rs_xtrees.best_score_
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed:    4.1s
[Parallel(n_jobs=-1)]: Done 146 tasks      | elapsed:   21.5s
[Parallel(n_jobs=-1)]: Done 349 tasks      | elapsed:   43.2s
[Parallel(n jobs=-1)]: Done 500 out of 500 | elapsed:   59.3s finished
Wall time: 59.8 s
```

Out[60]:

```
({'n_estimators': 200,
  'min_samples_split': 2,
  'min_samples_leaf': 1,
  'max_features': 0.5,
  'max_depth': None},
 0.8989256198347106)
```

5.3 Description of best extra trees classifier

In [61]:

```
# Get the best parameter list
print('The parameter list for the best estimator')
best_xtrees = rs_xtrees.best_estimator_
pprint(best_xtrees.get_params())
```

```
The parameter list for the best estimator
{'bootstrap': False,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 0.5,
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_impurity_split': None,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 200,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 33,
 'verbose': 0,
 'warm start': False}
```

5.4 Analysis the performance of the best estimator

In [62]:

```
print('Training accuracy: {:.2f}%'.format(
    best_xtrees.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_xtrees.score(X_important_test, y_test) * 100))
```

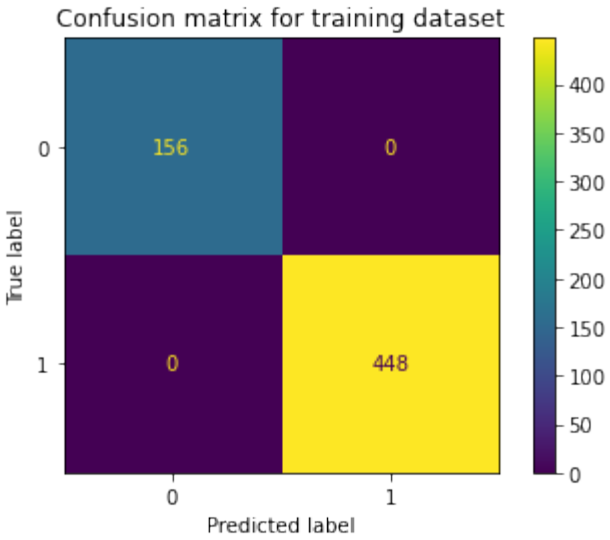
```
Training accuracy: 100.00%
Test accuracy:     91.45%
```

In [63]:

```
print('Performance on training data:\n')
```

```
print(
    classification_report(y_train,
                          best_xtrees.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_xtrees, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

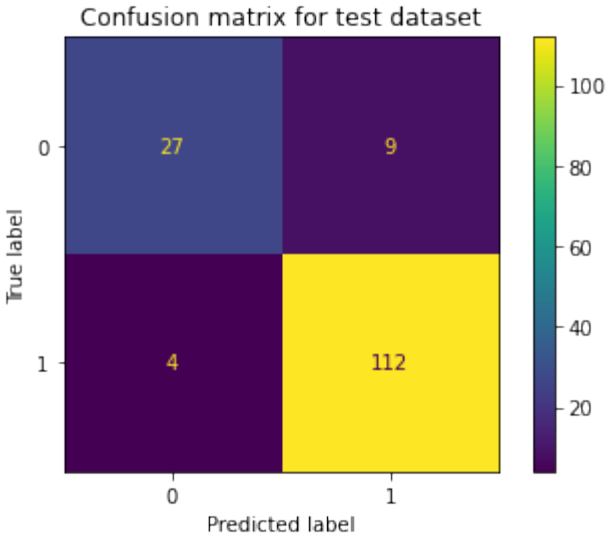
```
Performance on training data:

              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000       156
           1     1.0000    1.0000    1.0000       448

    accuracy                         1.0000       604
   macro avg     1.0000    1.0000    1.0000       604
weighted avg     1.0000    1.0000    1.0000       604
```



In [64]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          best_xtrees.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_xtrees, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:

              precision    recall  f1-score   support

           0     0.8710    0.7500    0.8060        36
           1     0.9256    0.9655    0.9451       116

    accuracy                         0.9145       152
   macro avg     0.8983    0.8578    0.8756       152
weighted avg     0.9127    0.9145    0.9122       152
```



6 Logistic Regression Tuning using Grid Hyperparameter Search

6.1 Set up parameter grid

In [65]:

```
param_grid = {
    'penalty': ['l1', 'l2', 'elasticnet'],
    'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
    'C': np.logspace(-5, 0, num=6)
}

pprint(param_grid)
```

```
{'C': array([1.e-05, 1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00]),
 'penalty': ['l1', 'l2', 'elasticnet'],
 'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']}
```

6.2 Grid hypterparameter search training

In [66]:

```
lr_model = LogisticRegression(max_iter=10000, random_state=RANDOM_SEED)

gs_lr = GridSearchCV(lr_model,
                     param_grid=param_grid,
                     cv=5,
                     verbose=2,
                     n_jobs=-1)

gs_lr.fit(X_important_train, y_train)
gs_lr.best_params_, gs_lr.best_score_
```

```
Fitting 5 folds for each of 90 candidates, totalling 450 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:    0.0s
[Parallel(n_jobs=-1)]: Done 450 out of 450 | elapsed:   12.4s finished
```

Out[66]:

```
({'C': 0.1, 'penalty': 'l1', 'solver': 'liblinear'}, 0.8558953168044077)
```

6.3 Patameter list for the best LogisticRegression model

In [67]:

```
best_lr = gs_lr.best_estimator_
print('Paramater list for the estimator')
pprint(best_lr.get_params())
```

```
Paramater list for the estimator
{'C': 0.1,
 'class_weight': None,
 'dual': False,
 'fit_intercept': True,
 'intercept_scaling': 1,
 'l1_ratio': None,
 'max_iter': 10000,
 'multi_class': 'auto',
 'n_jobs': None,
 'penalty': 'l1',
 'random_state': 33,
 'solver': 'liblinear',
 'tol': 0.0001,
 'verbose': 0,
 'warm start': False}
```

6.4 Analysis the performance of the best estimator

In [68]:

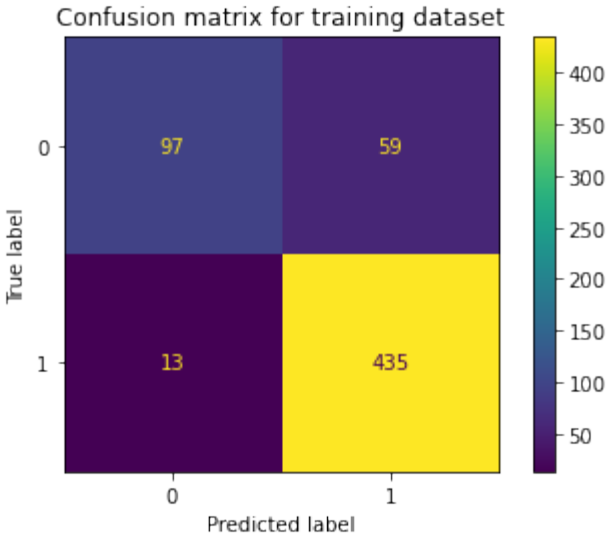
```
print('Training accuracy: {:.2f}%'.format(
    best_lr.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_lr.score(X_important_test, y_test) * 100))
```

```
Training accuracy: 88.08%
Test accuracy:     91.45%
```

In [69]:

```
print('Performance on training data:\n')
print(
    classification_report(y_train,
                          best_lr.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_lr, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

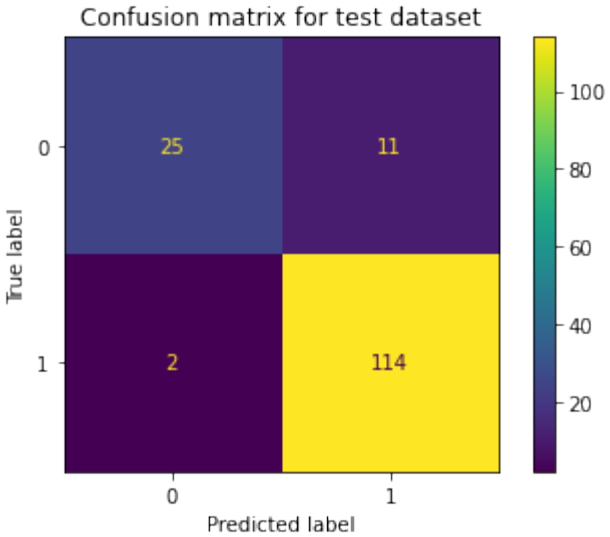
```
Performance on training data:

              precision    recall  f1-score   support

           0     0.8818    0.6218    0.7293       156
           1     0.8806    0.9710    0.9236       448

    accuracy                         0.8808       604
   macro avg     0.8812    0.7964    0.8264       604
weighted avg     0.8809    0.8808    0.8734       604
```



In [70]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          best_lr.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_lr, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:

              precision    recall  f1-score   support

           0     0.9259    0.6944    0.7937        36
           1     0.9120    0.9828    0.9461       116

    accuracy                         0.9145       152
   macro avg     0.9190    0.8386    0.8699       152
weighted avg     0.9153    0.9145    0.9100       152
```



7 SVM Tuning using Random Hyperparameter Search

7.1 Set up parameter distribution

In [71]:

```
param_grid = {
    'C': np.logspace(-2, 3, num=6),
    'gamma': np.logspace(-4, 0, num=5),
    'kernel': ['rbf', 'poly', 'sigmoid', 'linear']
}

pprint(param_grid)
```

```
{'C': array([1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03]),
 'gamma': array([1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00]),
 'kernel': ['rbf', 'poly', 'sigmoid', 'linear']}
```

7.2 Grid hypterparameter search training

In [72]:

```
svm_model = SVC(max_iter=10000, random_state=RANDOM_SEED)

gs_svm = GridSearchCV(svm_model,
                      param_grid=param_grid,
                      cv=5,
                      verbose=2,
                      n_jobs=-1)

gs_svm.fit(X_important_train, y_train)
gs_svm.best_params_, gs_svm.best_score_
```

```
Fitting 5 folds for each of 120 candidates, totalling 600 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:    0.2s
[Parallel(n_jobs=-1)]: Done 504 tasks      | elapsed:    3.8s
[Parallel(n_jobs=-1)]: Done 585 out of 600 | elapsed:    4.4s remaining:    0.0s
[Parallel(n jobs=-1)]: Done 600 out of 600 | elapsed:    4.5s finished
```

Out[72]:

```
({'C': 100.0, 'gamma': 0.001, 'kernel': 'rbf'}, 0.889090909090909)
```

7.3 Patameter list for the best SVM

In [73]:

```
best_svm = gs_svm.best_estimator_
print('Paramater list for the estimator')
pprint(best_svm.get_params())
```

```
Paramater list for the estimator
{'C': 100.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': None,
 'coef0': 0.0,
 'decision_function_shape': 'ovr',
 'degree': 3,
 'gamma': 0.001,
 'kernel': 'rbf',
 'max_iter': 10000,
 'probability': False,
 'random_state': 33,
 'shrinking': True,
 'tol': 0.001,
 'verbose': False}
```

7.4 Analysis the performance of the best estimator

In [74]:

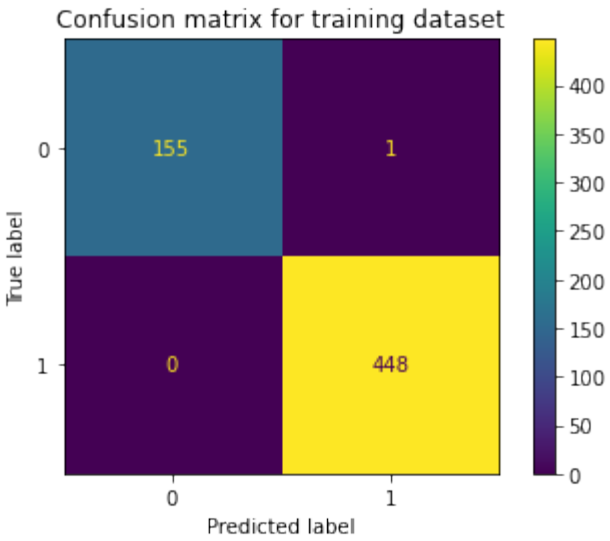
```
print('Training accuracy: {:.2f}%'.format(
    best_svm.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_svm.score(X_important_test, y_test) * 100))
```

```
Training accuracy: 99.83%
Test accuracy:     92.76%
```

In [75]:

```
print('Performance on training data:\n')
print(
    classification_report(y_train,
                          best_svm.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_svm, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

```
Performance on training data:

              precision    recall  f1-score   support

           0     1.0000    0.9936    0.9968       156
           1     0.9978    1.0000    0.9989       448

    accuracy                         0.9983       604
   macro avg     0.9989    0.9968    0.9978       604
weighted avg     0.9983    0.9983    0.9983       604
```

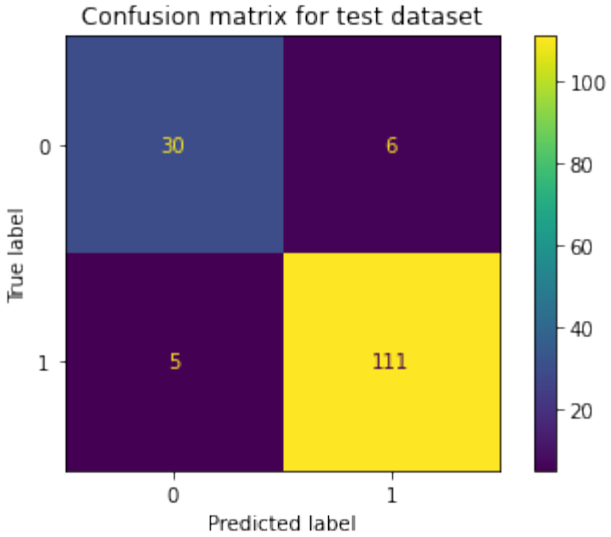


In [76]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          best_svm.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_svm, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:
```

```
              precision    recall  f1-score   support

           0     0.8571    0.8333    0.8451        36
           1     0.9487    0.9569    0.9528       116

    accuracy                         0.9276       152
   macro avg     0.9029    0.8951    0.8989       152
weighted avg     0.9270    0.9276    0.9273       152
```



8 XGBoost Tuning using Grid Hyperparameter Search

8.1 Find the parameters for the best estimator

8.1.1 Find best max_depth and max_depth

In [77]:

```
param_grid1 = {'max_depth': [7, 8, 9], 'min_child_weight': [2, 4, 6, 8]}

pprint(param_grid1)
```

```
{'max depth': [7, 8, 9], 'min child weight': [2, 4, 6, 8]}
```

In [78]:

```
xgb_clf1 = xgb.XGBClassifier(learning_rate=0.05,
                             n_estimators=10000,
                             gamma=0,
                             objective='binary:logistic',
                             nthread=4,
                             scale_pos_weight=1,
                             seed=RANDOM_SEED)

gs_xgb1 = GridSearchCV(
    xgb_clf1,
    param_grid=param_grid1,
    verbose=2,
    cv=5,
    n_jobs=-1,
)

gs_xgb1.fit(X_important_train, y_train)
gs_xgb1.best_params_, gs_xgb1.best_score_
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed:  2.0min
[Parallel(n jobs=-1)]: Done  60 out of  60 | elapsed:  3.8min finished
```

Out[78]:

```
({'max depth': 8, 'min child weight': 2}, 0.8973278236914602)
```

In [79]:

```
best_xgb1 = gs_xgb1.best_estimator_
print('Training accuracy: {:.2f}%'.format(
    best_xgb1.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_xgb1.score(X_important_test, y_test) * 100))
```

```
Training accuracy: 100.00%
Test accuracy:     92.11%
```

8.1.2 Tune subsample and colsample_bytree

In [80]:

```
param_grid2 = {
    'subsample': np.arange(0.6, 1, 0.1),
    'colsample_bytree': np.arange(0.6, 1, 0.1)
}

pprint(param_grid2)
```

```
{'colsample_bytree': array([0.6, 0.7, 0.8, 0.9]),
 'subsample': array([0.6, 0.7, 0.8, 0.9])}
```

In [81]:

```
xgb_clf2 = xgb.XGBClassifier(max_depth=8,
                             min_child_weight=2,
                             learning_rate=0.05,
                             n_estimators=10000,
                             gamma=0,
                             objective='binary:logistic',
                             nthread=4,
                             booster='gbtree',
                             scale_pos_weight=1,
                             seed=RANDOM_SEED)

gs_xgb2 = GridSearchCV(
    xgb_clf2,
    param_grid=param_grid2,
    verbose=2,
    cv=5,
    n_jobs=-1,
)

gs_xgb2.fit(X_important_train, y_train)
gs_xgb2.best_params_, gs_xgb2.best_score_
```

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed:  1.7min
[Parallel(n jobs=-1)]: Done  80 out of  80 | elapsed:  4.6min finished
```

Out[81]:

```
({'colsample_bytree': 0.8999999999999999, 'subsample': 0.6},
 0.9089118457300277)
```

In [82]:

```
best_xgb2 = gs_xgb2.best_estimator_
```

```
print('Training accuracy: {:.2f}%'.format(
    best_xgb2.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_xgb2.score(X_important_test, y_test) * 100))
```

```
Training accuracy: 100.00%
Test accuracy:     90.79%
```

8.1.3 Tune learning rate and number of gradient boosted trees

In [83]:

```
param_grid3 = {
    'learning_rate': [0.1, 0.15, 0.2],
    'n_estimators': [400, 600, 800, 1000]
}
pprint(param_grid3)
```

```
{'learning rate': [0.1, 0.15, 0.2], 'n estimators': [400, 600, 800, 1000]}
```

In [84]:

```
xgb_clf3 = xgb.XGBClassifier(max_depth=4,
                             min_child_weight=2,
                             gamma=0,
                             objective='binary:logistic',
                             colsample_bytree=0.6,
                             subsample=0.9,
                             nthread=4,
                             scale_pos_weight=1,
                             seed=27)

gs_xgb3 = GridSearchCV(
    xgb_clf3,
    param_grid=param_grid3,
    verbose=2,
    cv=5,
    n_jobs=-1,
)

gs_xgb3.fit(X_important_train, y_train)
gs_xgb3.best_params_, gs_xgb3.best_score_
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed:   10.3s
[Parallel(n jobs=-1)]: Done  60 out of  60 | elapsed:   20.8s finished
```

Out[84]:

```
({'learning rate': 0.1, 'n estimators': 400}, 0.9006473829201103)
```

8.2 Patameter list for the best XGBoost Classifier

In [85]:

```
best_xgb = gs_xgb3.best_estimator_
print('Paramater list for the estimator')
pprint(best_xgb.get_params())
```

```
Paramater list for the estimator
{'base_score': 0.5,
 'booster': 'gbtree',
 'colsample_bylevel': 1,
 'colsample_bynode': 1,
 'colsample_bytree': 0.6,
 'gamma': 0,
```

```
 'gpu_id': -1,
 'importance_type': 'gain',
 'interaction_constraints': '',
 'learning_rate': 0.1,
 'max_delta_step': 0,
 'max_depth': 4,
 'min_child_weight': 2,
 'missing': nan,
 'monotone_constraints': '()',
 'n_estimators': 400,
 'n_jobs': 4,
 'nthread': 4,
 'num_parallel_tree': 1,
 'objective': 'binary:logistic',
 'random_state': 27,
 'reg_alpha': 0,
 'reg_lambda': 1,
 'scale_pos_weight': 1,
 'seed': 27,
 'subsample': 0.9,
 'tree_method': 'exact',
 'validate_parameters': 1,
 'verbosity': None}
```

8.3 Analysis the performance of the best estimator

In [86]:

```
best_xgb.fit(X_important_train, y_train, verbose=2)
```

Out[86]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.6, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.1, max_delta_step=0, max_depth=4,
              min_child_weight=2, missing=nan, monotone_constraints='()',
              n_estimators=400, n_jobs=4, nthread=4, num_parallel_tree=1,
              random_state=27, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
              seed=27, subsample=0.9, tree_method='exact',
              validate parameters=1, verbosity=None)
```

In [87]:

```
print('Training accuracy: {:.2f}%'.format(
    best_xgb.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_xgb.score(X_important_test, y_test) * 100))
```

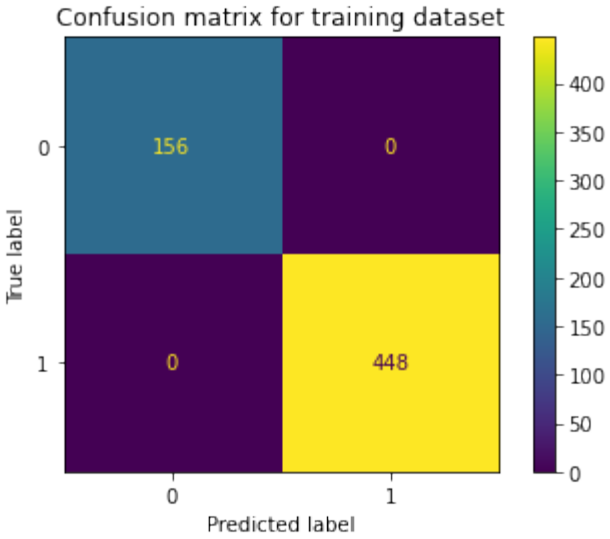
```
Training accuracy: 100.00%
Test accuracy:     93.42%
```

In [88]:

```
print('Performance on training data:\n')
print(
    classification_report(y_train,
                          best_xgb.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_xgb, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

```
Performance on training data:

              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000       156
           1     1.0000    1.0000    1.0000       448
```

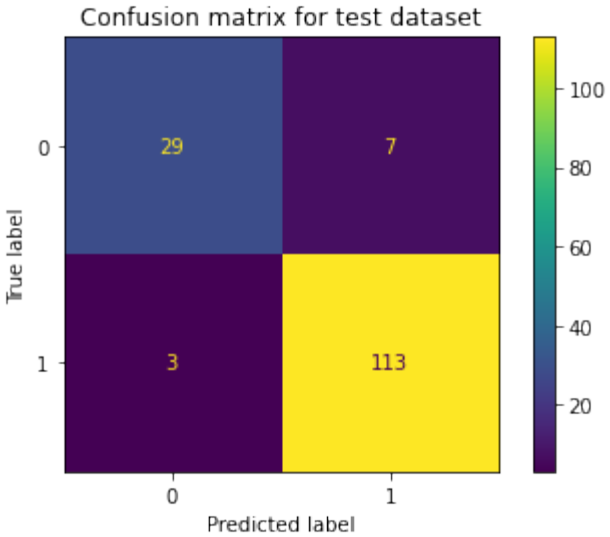
```
    accuracy                           1.0000       604
   macro avg     1.0000    1.0000    1.0000       604
weighted avg     1.0000    1.0000    1.0000       604
```



In [89]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          best_xgb.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_xgb, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:

              precision    recall  f1-score   support

           0     0.9062    0.8056    0.8529        36
           1     0.9417    0.9741    0.9576       116

    accuracy                         0.9342       152
   macro avg     0.9240    0.8898    0.9053       152
weighted avg     0.9333    0.9342    0.9328       152
```



9 CatBoost Classifier Tuning

9.1 Set up parameter distribution

In [90]:

```
param_grid = {'depth': np.arange(6, 9, 1), 'l2_leaf_reg': [1, 3, 5]}

pprint(param_grid)
```

```
{'depth': array([6, 7, 8]), 'l2 leaf reg': [1, 3, 5]}
```

9.2 Grid hypterparameter search training

In [91]:

```
catb_model = ctb.CatBoostClassifier(iterations=400,
                                    learning_rate=0.1,
                                    random_seed=RANDOM_SEED)

grid_search_result = catb_model.grid_search(param_grid,
                                            X=X_important_train,
                                            y=y_train,
                                            verbose=2,
                                            cv=5,
                                            partition_random_seed=RANDOM_SEED,
                                            plot=True)
```

```
bestTest = 0.2537038718
bestIteration = 125

0:      loss: 0.2537039 best: 0.2537039 (0)     total: 9.36s    remaining: 1m 14s

bestTest = 0.2260130241
bestIteration = 116


bestTest = 0.2376994556
bestIteration = 202

2:      loss: 0.2376995 best: 0.2260130 (1)     total: 27.8s    remaining: 55.7s

bestTest = 0.2548930323
bestIteration = 75


bestTest = 0.2605842385
bestIteration = 95

4:      loss: 0.2605842 best: 0.2260130 (1)     total: 1m 2s    remaining: 50s

bestTest = 0.2392508007
bestIteration = 167


bestTest = 0.2138337157
bestIteration = 70

6:      loss: 0.2138337 best: 0.2138337 (6)     total: 1m 54s   remaining: 32.6s

bestTest = 0.2704721113
bestIteration = 94


bestTest = 0.2440361871
bestIteration = 122
```

```
8:      loss: 0.2440362 best: 0.2138337 (6)     total: 3m 2s    remaining: 0us
Estimating final quality...
```

9.3 Patameter list for the best CatBoost Classifier

In [92]:

```
best_ctb = ctb.CatBoostClassifier(random_seed=RANDOM_SEED,
                                  **grid_search_result['params'])
best_ctb.fit(X_important_train, y_train, logging_level='Silent')
```

Out[92]:

```
<catboost.core.CatBoostClassifier at 0x19eb6fddc40>
```

In [93]:

```
best_ctb.get_params()
```

Out[93]:

```
{'depth': 8, 'l2 leaf reg': 1, 'random seed': 33}
```

9.4 Analysis the performance of the best estimator

In [94]:

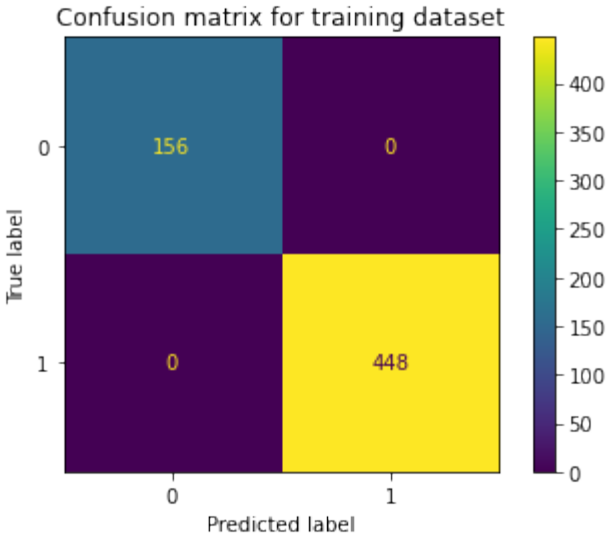
```
print('Training accuracy: {:.2f}%'.format(
    best_ctb.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_ctb.score(X_important_test, y_test) * 100))
```

```
Training accuracy: 100.00%
Test accuracy:     93.42%
```

In [95]:

```
print('Performance on training data:\n')
print(
    classification_report(y_train,
                          best_ctb.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_ctb, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

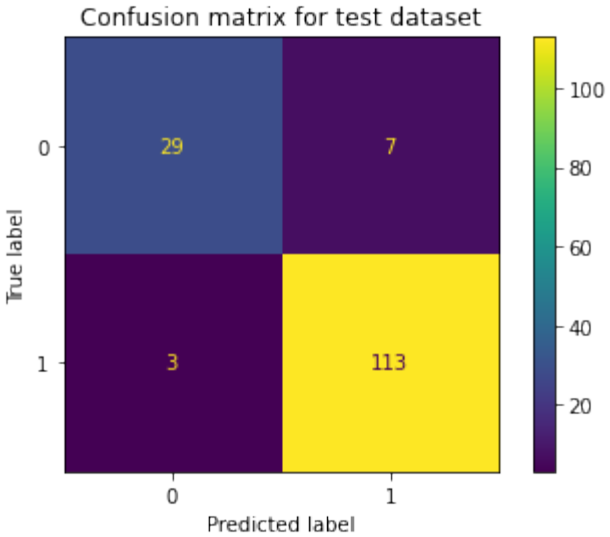
```
Performance on training data:

              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000       156
           1     1.0000    1.0000    1.0000       448

    accuracy                         1.0000       604
   macro avg     1.0000    1.0000    1.0000       604
weighted avg     1.0000    1.0000    1.0000       604
```



In [96]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          best_ctb.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_ctb, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:

              precision    recall  f1-score   support

           0     0.9062    0.8056    0.8529        36
           1     0.9417    0.9741    0.9576       116

    accuracy                         0.9342       152
   macro avg     0.9240    0.8898    0.9053       152
weighted avg     0.9333    0.9342    0.9328       152
```



10 LGBMClassifier Tuning using Bayes Hyperparameter Search

10.1 Set up search spaces

In [97]:

```
search_spaces = {
    'num_leaves': Integer(10, 40),
    'max_depth': Integer(4, 10),
    'boosting_type': Categorical(['gbdt', 'dart']),
    'colsample_bytree': Real(0.8, 1),
    'subsample': Real(0.8, 1),
}

pprint(search_spaces)
```

```
{'boosting_type': Categorical(categories=('gbdt', 'dart'), prior=None),
 'colsample_bytree': Real(low=0.8, high=1, prior='uniform', transform='identity'),
 'max_depth': Integer(low=4, high=10, prior='uniform', transform='identity'),
 'num_leaves': Integer(low=10, high=40, prior='uniform', transform='identity'),
 'subsample': Real(low=0.8, high=1, prior='uniform', transform='identity')}
```

10.2 Bayes hypterparameter search training

In [98]:

```
lgbm_model = lgb.LGBMClassifier(
    learning_rate=0.1,
    n_estimators=400,
    objective='binary',
    subsample_for_bin=200,
    subsample=0.8,
    subsample_freq=1,
    min_split_gain=0.5,
    min_child_weight=1,
    min_child_samples=5,
    scale_pos_weight=1,
    n_jobs=-1,
    random_state=RANDOM_SEED,
    silent=True,
)

bs_lgbm = BayesSearchCV(lgbm_model,
                        search_spaces=search_spaces,
                        n_iter=300,
                        cv=5,
                        random_state=RANDOM_SEED,
                        n_jobs=-1,
                        verbose=2)

bs_lgbm.fit(X_important_train, y_train)

bs_lgbm.best_score_, bs_lgbm.best_params_
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    7.3s remaining:   11.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    7.3s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    7.3s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    1.4s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    1.4s finished
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
```

```
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.1s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.7s remaining:    1.1s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.7s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
```

```
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
```

```
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.7s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.7s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
```

```
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    0.9s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.7s remaining:    1.1s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.3s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.7s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.7s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.5s remaining:    0.8s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.5s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.2s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.2s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.6s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.6s finished
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:    0.7s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:    0.7s finished
```

Out[98]:

```
(0.9122516556291391,
 OrderedDict([('boosting_type', 'dart'),
              ('colsample_bytree', 0.8016779670151458),
              ('max_depth', 10),
              ('num_leaves', 26),
              ('subsample', 0.8012317612884369)]))
```

10.3 Patameter list for the best LightGBM classifier

In [99]:

```
best_lgbm = bs_lgbm.best_estimator_
print('Paramater list for the estimator')
pprint(best_lgbm.get_params())
```

```
Paramater list for the estimator
{'boosting_type': 'dart',
 'class_weight': None,
 'colsample_bytree': 0.8016779670151458,
 'importance_type': 'split',
 'learning_rate': 0.1,
 'max_depth': 10,
 'min_child_samples': 5,
 'min_child_weight': 1,
 'min_split_gain': 0.5,
 'n_estimators': 400,
 'n_jobs': -1,
```

```
 'num_leaves': 26,
 'objective': 'binary',
 'random_state': 33,
 'reg_alpha': 0.0,
 'reg_lambda': 0.0,
 'scale_pos_weight': 1,
 'silent': True,
 'subsample': 0.8012317612884369,
 'subsample_for_bin': 200,
 'subsample freq': 1}
```

10.4 Analysis the performance of the best estimator

In [100]:

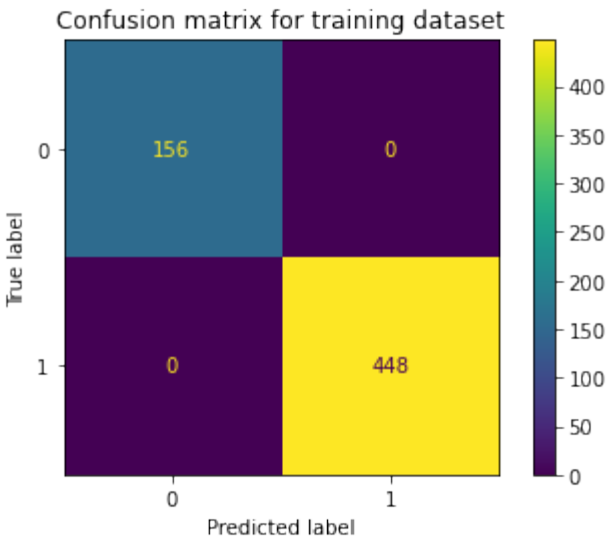
```
print('Training accuracy: {:.2f}%'.format(
    best_lgbm.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_lgbm.score(X_important_test, y_test) * 100))
```

```
Training accuracy: 100.00%
Test accuracy:     93.42%
```

In [101]:

```
print('Performance on training data:\n')
print(
    classification_report(y_train,
                          best_lgbm.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_lgbm, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

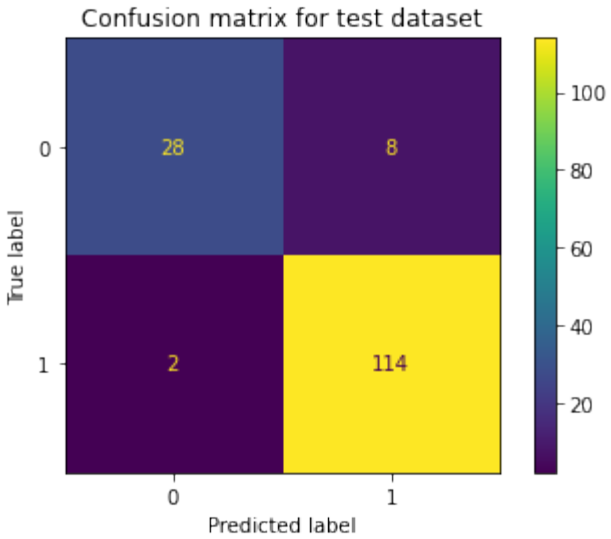
```
Performance on training data:

              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000       156
           1     1.0000    1.0000    1.0000       448

    accuracy                         1.0000       604
   macro avg     1.0000    1.0000    1.0000       604
weighted avg     1.0000    1.0000    1.0000       604
```



In [102]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          best_lgbm.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_lgbm, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:

              precision    recall  f1-score   support

           0     0.9333    0.7778    0.8485        36
           1     0.9344    0.9828    0.9580       116

    accuracy                         0.9342       152
   macro avg     0.9339    0.8803    0.9032       152
weighted avg     0.9342    0.9342    0.9320       152
```



11 K-nearest Neighbor Classifier Tuning using Grid Hyperparameter Search

11.1 Set up search spaces

In [103]:

```
param_grid = {'n_neighbors': np.arange(1, 6), 'p': [1, 2, 3]}

pprint(param_grid)
```

```
{'n neighbors': array([1, 2, 3, 4, 5]), 'p': [1, 2, 3]}
```

11.2 Bayes hypterparameter search training

In [104]:

```
knn_clf = KNeighborsClassifier(algorithm='auto', n_jobs=-1)

gs_knn = GridSearchCV(
    knn_clf,
    param_grid=param_grid,
    verbose=2,
    cv=5,
```

```
    n_jobs=-1,
)

gs_knn.fit(X_important_train, y_train)
gs_knn.best_params_, gs_knn.best_score_
```

```
Fitting 5 folds for each of 15 candidates, totalling 75 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:    1.1s
[Parallel(n_jobs=-1)]: Done  60 out of  75 | elapsed:    1.9s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done  75 out of  75 | elapsed:    2.4s finished
```

Out[104]:

```
({'n_neighbors': 1, 'p': 1}, 0.941969696969697)
```

11.3 Patameter list for the best KNN classifier

In [105]:

```
best_knn = gs_knn.best_estimator_
print('Paramater list for the estimator')
pprint(best_knn.get_params())
```

```
Paramater list for the estimator
{'algorithm': 'auto',
 'leaf_size': 30,
 'metric': 'minkowski',
 'metric_params': None,
 'n_jobs': -1,
 'n_neighbors': 1,
 'p': 1,
 'weights': 'uniform'}
```

11.4 Analysis the performance of the best estimator

In [106]:

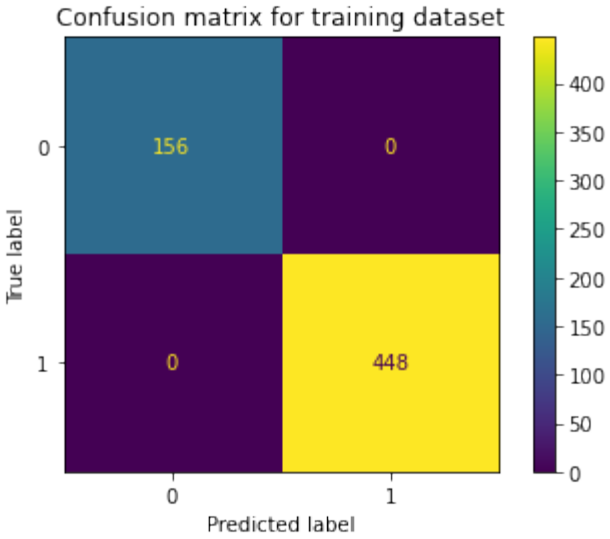
```
print('Training accuracy: {:.2f}%'.format(
    best_knn.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    best_knn.score(X_important_test, y_test) * 100))
```

```
Training accuracy: 100.00%
Test accuracy:     97.37%
```

In [107]:

```
print('Performance on training data:\n')
print(
    classification_report(y_train,
                          best_knn.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_knn, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

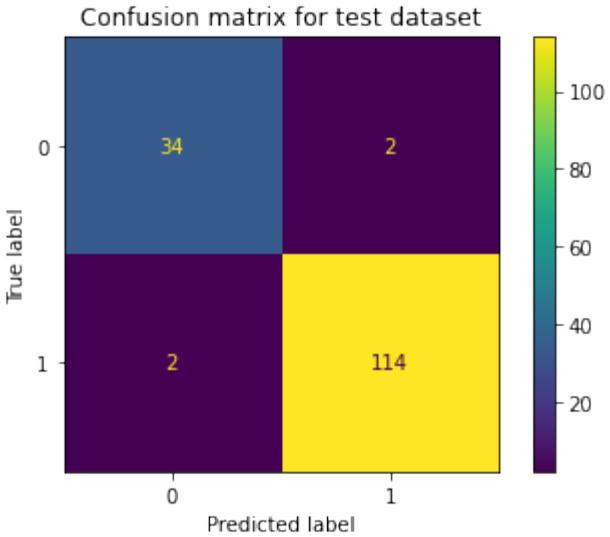
```
Performance on training data:

              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000       156
           1     1.0000    1.0000    1.0000       448

    accuracy                         1.0000       604
   macro avg     1.0000    1.0000    1.0000       604
weighted avg     1.0000    1.0000    1.0000       604
```



In [108]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          best_knn.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(best_knn, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:

              precision    recall  f1-score   support

           0     0.9444    0.9444    0.9444        36
           1     0.9828    0.9828    0.9828       116

    accuracy                         0.9737       152
   macro avg     0.9636    0.9636    0.9636       152
weighted avg     0.9737    0.9737    0.9737       152
```



12 Combine Estimators by StakcingCVClassifier

The StackingCVClassifier extends the standard stacking algorithm (implemented as StackingClassifier) using cross-validation to prepare the input data for the level-2 classifier.

12.1 Collect the estimators

In [157]:

```
import warnings
warnings.simplefilter('ignore')

estimators = [best_xtrees, best_svm, best_xgb, best_ctb, best_lgbm, best_knn]

est_names = [
    'Extra Trees', 'SVM', 'XGBoost Classifier', 'CatBoost Classifier',
    'LightGBM Classifier', 'KNN', 'StackingCVClassifier'
]
```

12.2 Training the data

In [136]:

```
sclf = StackingCVClassifier(classifiers=estimators,
                            meta_classifier=best_lr,
                            cv=5,
                            n_jobs=-1,
                            verbose=0,
                            random_state=RANDOM_SEED
                           )

sclf.fit(X_important_train, y_train)
```

```
0:      learn: 0.6665931        total: 112ms    remaining: 1m 52s
1:      learn: 0.6369221        total: 199ms    remaining: 1m 39s
2:      learn: 0.6121429        total: 287ms    remaining: 1m 35s
3:      learn: 0.5893073        total: 377ms    remaining: 1m 33s
4:      learn: 0.5652527        total: 472ms    remaining: 1m 33s
5:      learn: 0.5444878        total: 558ms    remaining: 1m 32s
6:      learn: 0.5275610        total: 652ms    remaining: 1m 32s
7:      learn: 0.5080990        total: 744ms    remaining: 1m 32s
8:      learn: 0.4885503        total: 831ms    remaining: 1m 31s
9:      learn: 0.4732062        total: 931ms    remaining: 1m 32s
10:     learn: 0.4574211        total: 1.02s    remaining: 1m 31s
11:     learn: 0.4416857        total: 1.11s    remaining: 1m 31s
12:     learn: 0.4278373        total: 1.2s     remaining: 1m 31s
13:     learn: 0.4137917        total: 1.3s     remaining: 1m 31s
14:     learn: 0.4023856        total: 1.4s     remaining: 1m 31s
15:     learn: 0.3918084        total: 1.5s     remaining: 1m 32s
16:     learn: 0.3808521        total: 1.58s    remaining: 1m 31s
17:     learn: 0.3699718        total: 1.68s    remaining: 1m 31s
18:     learn: 0.3571421        total: 1.76s    remaining: 1m 31s
19:     learn: 0.3466100        total: 1.86s    remaining: 1m 31s
20:     learn: 0.3359778        total: 1.95s    remaining: 1m 30s
21:     learn: 0.3260458        total: 2.03s    remaining: 1m 30s
22:     learn: 0.3160314        total: 2.12s    remaining: 1m 30s
23:     learn: 0.3076384        total: 2.21s    remaining: 1m 29s
24:     learn: 0.2973433        total: 2.31s    remaining: 1m 29s
25:     learn: 0.2910566        total: 2.4s     remaining: 1m 29s
26:     learn: 0.2817621        total: 2.49s    remaining: 1m 29s
27:     learn: 0.2734052        total: 2.59s    remaining: 1m 29s
28:     learn: 0.2658563        total: 2.67s    remaining: 1m 29s
29:     learn: 0.2593366        total: 2.77s    remaining: 1m 29s
30:     learn: 0.2511561        total: 2.87s    remaining: 1m 29s
31:     learn: 0.2436404        total: 2.97s    remaining: 1m 29s
32:     learn: 0.2378629        total: 3.07s    remaining: 1m 29s
33:     learn: 0.2328022        total: 3.15s    remaining: 1m 29s
```

```
34:     learn: 0.2268950        total: 3.25s    remaining: 1m 29s
35:     learn: 0.2205209        total: 3.34s    remaining: 1m 29s
36:     learn: 0.2148865        total: 3.44s    remaining: 1m 29s
37:     learn: 0.2109698        total: 3.53s    remaining: 1m 29s
38:     learn: 0.2061818        total: 3.63s    remaining: 1m 29s
39:     learn: 0.2002269        total: 3.72s    remaining: 1m 29s
40:     learn: 0.1945628        total: 3.82s    remaining: 1m 29s
41:     learn: 0.1901853        total: 3.91s    remaining: 1m 29s
42:     learn: 0.1855108        total: 4s       remaining: 1m 29s
43:     learn: 0.1813334        total: 4.09s    remaining: 1m 28s
44:     learn: 0.1774749        total: 4.17s    remaining: 1m 28s
45:     learn: 0.1742857        total: 4.26s    remaining: 1m 28s
46:     learn: 0.1709442        total: 4.35s    remaining: 1m 28s
47:     learn: 0.1672066        total: 4.44s    remaining: 1m 28s
48:     learn: 0.1637110        total: 4.53s    remaining: 1m 27s
49:     learn: 0.1610062        total: 4.62s    remaining: 1m 27s
50:     learn: 0.1574836        total: 4.7s     remaining: 1m 27s
51:     learn: 0.1553096        total: 4.8s     remaining: 1m 27s
52:     learn: 0.1526120        total: 4.89s    remaining: 1m 27s
53:     learn: 0.1505715        total: 4.99s    remaining: 1m 27s
54:     learn: 0.1469025        total: 5.07s    remaining: 1m 27s
55:     learn: 0.1442962        total: 5.16s    remaining: 1m 26s
56:     learn: 0.1410970        total: 5.25s    remaining: 1m 26s
57:     learn: 0.1382398        total: 5.34s    remaining: 1m 26s
58:     learn: 0.1354866        total: 5.43s    remaining: 1m 26s
59:     learn: 0.1334406        total: 5.52s    remaining: 1m 26s
60:     learn: 0.1311775        total: 5.61s    remaining: 1m 26s
61:     learn: 0.1291246        total: 5.71s    remaining: 1m 26s
62:     learn: 0.1267156        total: 5.79s    remaining: 1m 26s
63:     learn: 0.1238921        total: 5.89s    remaining: 1m 26s
64:     learn: 0.1217356        total: 5.98s    remaining: 1m 26s
65:     learn: 0.1195336        total: 6.07s    remaining: 1m 25s
66:     learn: 0.1175291        total: 6.16s    remaining: 1m 25s
67:     learn: 0.1149171        total: 6.25s    remaining: 1m 25s
68:     learn: 0.1131604        total: 6.33s    remaining: 1m 25s
69:     learn: 0.1109330        total: 6.42s    remaining: 1m 25s
70:     learn: 0.1090313        total: 6.51s    remaining: 1m 25s
71:     learn: 0.1076120        total: 6.6s     remaining: 1m 25s
72:     learn: 0.1055416        total: 6.69s    remaining: 1m 24s
73:     learn: 0.1038735        total: 6.78s    remaining: 1m 24s
74:     learn: 0.1022902        total: 6.87s    remaining: 1m 24s
75:     learn: 0.1006721        total: 6.96s    remaining: 1m 24s
76:     learn: 0.0987378        total: 7.05s    remaining: 1m 24s
77:     learn: 0.0973889        total: 7.14s    remaining: 1m 24s
78:     learn: 0.0954455        total: 7.22s    remaining: 1m 24s
79:     learn: 0.0936286        total: 7.31s    remaining: 1m 24s
80:     learn: 0.0923240        total: 7.39s    remaining: 1m 23s
81:     learn: 0.0911380        total: 7.48s    remaining: 1m 23s
82:     learn: 0.0892142        total: 7.57s    remaining: 1m 23s
83:     learn: 0.0881188        total: 7.66s    remaining: 1m 23s
84:     learn: 0.0872838        total: 7.74s    remaining: 1m 23s
85:     learn: 0.0860535        total: 7.83s    remaining: 1m 23s
86:     learn: 0.0848872        total: 7.92s    remaining: 1m 23s
87:     learn: 0.0839426        total: 8.01s    remaining: 1m 22s
88:     learn: 0.0822884        total: 8.1s     remaining: 1m 22s
89:     learn: 0.0808639        total: 8.19s    remaining: 1m 22s
90:     learn: 0.0800833        total: 8.27s    remaining: 1m 22s
91:     learn: 0.0791321        total: 8.36s    remaining: 1m 22s
92:     learn: 0.0780913        total: 8.45s    remaining: 1m 22s
93:     learn: 0.0769898        total: 8.54s    remaining: 1m 22s
94:     learn: 0.0758830        total: 8.63s    remaining: 1m 22s
95:     learn: 0.0749293        total: 8.72s    remaining: 1m 22s
96:     learn: 0.0738071        total: 8.81s    remaining: 1m 22s
97:     learn: 0.0723964        total: 8.91s    remaining: 1m 21s
98:     learn: 0.0711163        total: 8.99s    remaining: 1m 21s
99:     learn: 0.0702033        total: 9.09s    remaining: 1m 21s
100:    learn: 0.0692712        total: 9.18s    remaining: 1m 21s
101:    learn: 0.0682152        total: 9.28s    remaining: 1m 21s
102:    learn: 0.0671951        total: 9.36s    remaining: 1m 21s
```

```
103:    learn: 0.0661143        total: 9.46s    remaining: 1m 21s
104:    learn: 0.0651683        total: 9.54s    remaining: 1m 21s
105:    learn: 0.0645015        total: 9.63s    remaining: 1m 21s
106:    learn: 0.0635525        total: 9.72s    remaining: 1m 21s
107:    learn: 0.0629830        total: 9.82s    remaining: 1m 21s
108:    learn: 0.0619736        total: 9.91s    remaining: 1m 21s
109:    learn: 0.0610629        total: 10s      remaining: 1m 20s
110:    learn: 0.0600692        total: 10.1s    remaining: 1m 20s
111:    learn: 0.0594773        total: 10.2s    remaining: 1m 20s
112:    learn: 0.0586926        total: 10.3s    remaining: 1m 20s
113:    learn: 0.0579906        total: 10.4s    remaining: 1m 20s
114:    learn: 0.0574282        total: 10.5s    remaining: 1m 20s
115:    learn: 0.0567268        total: 10.6s    remaining: 1m 20s
116:    learn: 0.0559868        total: 10.6s    remaining: 1m 20s
117:    learn: 0.0552869        total: 10.7s    remaining: 1m 20s
118:    learn: 0.0545884        total: 10.8s    remaining: 1m 20s
119:    learn: 0.0539090        total: 10.9s    remaining: 1m 19s
120:    learn: 0.0528602        total: 11s      remaining: 1m 19s
121:    learn: 0.0522435        total: 11.1s    remaining: 1m 19s
122:    learn: 0.0513587        total: 11.2s    remaining: 1m 19s
123:    learn: 0.0509132        total: 11.3s    remaining: 1m 19s
124:    learn: 0.0505370        total: 11.3s    remaining: 1m 19s
125:    learn: 0.0499324        total: 11.4s    remaining: 1m 19s
126:    learn: 0.0493611        total: 11.5s    remaining: 1m 19s
127:    learn: 0.0486005        total: 11.6s    remaining: 1m 19s
128:    learn: 0.0482156        total: 11.7s    remaining: 1m 18s
129:    learn: 0.0478700        total: 11.8s    remaining: 1m 18s
130:    learn: 0.0473576        total: 11.9s    remaining: 1m 18s
131:    learn: 0.0469390        total: 12s      remaining: 1m 18s
132:    learn: 0.0461538        total: 12.1s    remaining: 1m 18s
133:    learn: 0.0451014        total: 12.1s    remaining: 1m 18s
134:    learn: 0.0444075        total: 12.2s    remaining: 1m 18s
135:    learn: 0.0439429        total: 12.3s    remaining: 1m 18s
136:    learn: 0.0433780        total: 12.4s    remaining: 1m 18s
137:    learn: 0.0430927        total: 12.5s    remaining: 1m 18s
138:    learn: 0.0425382        total: 12.6s    remaining: 1m 17s
139:    learn: 0.0422400        total: 12.7s    remaining: 1m 17s
140:    learn: 0.0417839        total: 12.8s    remaining: 1m 17s
141:    learn: 0.0413530        total: 12.9s    remaining: 1m 17s
142:    learn: 0.0406295        total: 12.9s    remaining: 1m 17s
143:    learn: 0.0401271        total: 13s      remaining: 1m 17s
144:    learn: 0.0396290        total: 13.1s    remaining: 1m 17s
145:    learn: 0.0390866        total: 13.2s    remaining: 1m 17s
146:    learn: 0.0387293        total: 13.3s    remaining: 1m 17s
147:    learn: 0.0381789        total: 13.4s    remaining: 1m 16s
148:    learn: 0.0376551        total: 13.5s    remaining: 1m 16s
149:    learn: 0.0372854        total: 13.6s    remaining: 1m 16s
150:    learn: 0.0368793        total: 13.6s    remaining: 1m 16s
151:    learn: 0.0363714        total: 13.7s    remaining: 1m 16s
152:    learn: 0.0358615        total: 13.8s    remaining: 1m 16s
153:    learn: 0.0354923        total: 13.9s    remaining: 1m 16s
154:    learn: 0.0351307        total: 14s      remaining: 1m 16s
155:    learn: 0.0346607        total: 14.1s    remaining: 1m 16s
156:    learn: 0.0342436        total: 14.2s    remaining: 1m 16s
157:    learn: 0.0338963        total: 14.3s    remaining: 1m 15s
158:    learn: 0.0335239        total: 14.3s    remaining: 1m 15s
159:    learn: 0.0332274        total: 14.4s    remaining: 1m 15s
160:    learn: 0.0330024        total: 14.5s    remaining: 1m 15s
161:    learn: 0.0327476        total: 14.6s    remaining: 1m 15s
162:    learn: 0.0323317        total: 14.7s    remaining: 1m 15s
163:    learn: 0.0320039        total: 14.8s    remaining: 1m 15s
164:    learn: 0.0317032        total: 14.9s    remaining: 1m 15s
165:    learn: 0.0313857        total: 14.9s    remaining: 1m 15s
166:    learn: 0.0309395        total: 15s      remaining: 1m 14s
167:    learn: 0.0306148        total: 15.1s    remaining: 1m 14s
168:    learn: 0.0301048        total: 15.2s    remaining: 1m 14s
169:    learn: 0.0298195        total: 15.3s    remaining: 1m 14s
170:    learn: 0.0295175        total: 15.4s    remaining: 1m 14s
171:    learn: 0.0292236        total: 15.5s    remaining: 1m 14s
```

```
172:    learn: 0.0288997        total: 15.6s    remaining: 1m 14s
173:    learn: 0.0285403        total: 15.7s    remaining: 1m 14s
174:    learn: 0.0281333        total: 15.7s    remaining: 1m 14s
175:    learn: 0.0278478        total: 15.8s    remaining: 1m 14s
176:    learn: 0.0275233        total: 15.9s    remaining: 1m 14s
177:    learn: 0.0272564        total: 16s      remaining: 1m 13s
178:    learn: 0.0269426        total: 16.1s    remaining: 1m 13s
179:    learn: 0.0266498        total: 16.2s    remaining: 1m 13s
180:    learn: 0.0264301        total: 16.3s    remaining: 1m 13s
181:    learn: 0.0262303        total: 16.4s    remaining: 1m 13s
182:    learn: 0.0259918        total: 16.4s    remaining: 1m 13s
183:    learn: 0.0257237        total: 16.5s    remaining: 1m 13s
184:    learn: 0.0254217        total: 16.6s    remaining: 1m 13s
185:    learn: 0.0251569        total: 16.7s    remaining: 1m 13s
186:    learn: 0.0248650        total: 16.8s    remaining: 1m 12s
187:    learn: 0.0245975        total: 16.9s    remaining: 1m 12s
188:    learn: 0.0243700        total: 17s      remaining: 1m 12s
189:    learn: 0.0241399        total: 17s      remaining: 1m 12s
190:    learn: 0.0239456        total: 17.1s    remaining: 1m 12s
191:    learn: 0.0237296        total: 17.2s    remaining: 1m 12s
192:    learn: 0.0234593        total: 17.3s    remaining: 1m 12s
193:    learn: 0.0231904        total: 17.4s    remaining: 1m 12s
194:    learn: 0.0229281        total: 17.5s    remaining: 1m 12s
195:    learn: 0.0227019        total: 17.6s    remaining: 1m 12s
196:    learn: 0.0224625        total: 17.7s    remaining: 1m 11s
197:    learn: 0.0222636        total: 17.7s    remaining: 1m 11s
198:    learn: 0.0220648        total: 17.8s    remaining: 1m 11s
199:    learn: 0.0218714        total: 17.9s    remaining: 1m 11s
200:    learn: 0.0215954        total: 18s      remaining: 1m 11s
201:    learn: 0.0214015        total: 18.1s    remaining: 1m 11s
202:    learn: 0.0212099        total: 18.2s    remaining: 1m 11s
203:    learn: 0.0210137        total: 18.3s    remaining: 1m 11s
204:    learn: 0.0208243        total: 18.4s    remaining: 1m 11s
205:    learn: 0.0205551        total: 18.4s    remaining: 1m 11s
206:    learn: 0.0203384        total: 18.5s    remaining: 1m 11s
207:    learn: 0.0202272        total: 18.6s    remaining: 1m 10s
208:    learn: 0.0200951        total: 18.7s    remaining: 1m 10s
209:    learn: 0.0198748        total: 18.8s    remaining: 1m 10s
210:    learn: 0.0197045        total: 18.9s    remaining: 1m 10s
211:    learn: 0.0195176        total: 19s      remaining: 1m 10s
212:    learn: 0.0193795        total: 19.1s    remaining: 1m 10s
213:    learn: 0.0192032        total: 19.2s    remaining: 1m 10s
214:    learn: 0.0190297        total: 19.3s    remaining: 1m 10s
215:    learn: 0.0188329        total: 19.3s    remaining: 1m 10s
216:    learn: 0.0187302        total: 19.4s    remaining: 1m 10s
217:    learn: 0.0185748        total: 19.5s    remaining: 1m 10s
218:    learn: 0.0183820        total: 19.6s    remaining: 1m 9s
219:    learn: 0.0181493        total: 19.7s    remaining: 1m 9s
220:    learn: 0.0180025        total: 19.8s    remaining: 1m 9s
221:    learn: 0.0178509        total: 19.9s    remaining: 1m 9s
222:    learn: 0.0177060        total: 20s      remaining: 1m 9s
223:    learn: 0.0174983        total: 20.1s    remaining: 1m 9s
224:    learn: 0.0173452        total: 20.1s    remaining: 1m 9s
225:    learn: 0.0171867        total: 20.2s    remaining: 1m 9s
226:    learn: 0.0169679        total: 20.3s    remaining: 1m 9s
227:    learn: 0.0167954        total: 20.4s    remaining: 1m 9s
228:    learn: 0.0165651        total: 20.5s    remaining: 1m 8s
229:    learn: 0.0164059        total: 20.6s    remaining: 1m 8s
230:    learn: 0.0162922        total: 20.7s    remaining: 1m 8s
231:    learn: 0.0161450        total: 20.8s    remaining: 1m 8s
232:    learn: 0.0159811        total: 20.8s    remaining: 1m 8s
233:    learn: 0.0158482        total: 20.9s    remaining: 1m 8s
234:    learn: 0.0156980        total: 21s      remaining: 1m 8s
235:    learn: 0.0155717        total: 21.1s    remaining: 1m 8s
236:    learn: 0.0154394        total: 21.2s    remaining: 1m 8s
237:    learn: 0.0153234        total: 21.3s    remaining: 1m 8s
238:    learn: 0.0151606        total: 21.4s    remaining: 1m 8s
239:    learn: 0.0149941        total: 21.5s    remaining: 1m 7s
240:    learn: 0.0148548        total: 21.6s    remaining: 1m 7s
```

```
241:    learn: 0.0147540        total: 21.6s    remaining: 1m 7s
242:    learn: 0.0146402        total: 21.7s    remaining: 1m 7s
243:    learn: 0.0145177        total: 21.8s    remaining: 1m 7s
244:    learn: 0.0144201        total: 21.9s    remaining: 1m 7s
245:    learn: 0.0142866        total: 22s      remaining: 1m 7s
246:    learn: 0.0141682        total: 22.1s    remaining: 1m 7s
247:    learn: 0.0140349        total: 22.2s    remaining: 1m 7s
248:    learn: 0.0138885        total: 22.2s    remaining: 1m 7s
249:    learn: 0.0137413        total: 22.3s    remaining: 1m 6s
250:    learn: 0.0136103        total: 22.4s    remaining: 1m 6s
251:    learn: 0.0134533        total: 22.5s    remaining: 1m 6s
252:    learn: 0.0133668        total: 22.6s    remaining: 1m 6s
253:    learn: 0.0132241        total: 22.7s    remaining: 1m 6s
254:    learn: 0.0131111        total: 22.8s    remaining: 1m 6s
255:    learn: 0.0130105        total: 22.9s    remaining: 1m 6s
256:    learn: 0.0128791        total: 22.9s    remaining: 1m 6s
257:    learn: 0.0127822        total: 23s      remaining: 1m 6s
258:    learn: 0.0126790        total: 23.1s    remaining: 1m 6s
259:    learn: 0.0125855        total: 23.2s    remaining: 1m 6s
260:    learn: 0.0124699        total: 23.3s    remaining: 1m 5s
261:    learn: 0.0123810        total: 23.4s    remaining: 1m 5s
262:    learn: 0.0123066        total: 23.5s    remaining: 1m 5s
263:    learn: 0.0121581        total: 23.5s    remaining: 1m 5s
264:    learn: 0.0120654        total: 23.6s    remaining: 1m 5s
265:    learn: 0.0119795        total: 23.7s    remaining: 1m 5s
266:    learn: 0.0118726        total: 23.8s    remaining: 1m 5s
267:    learn: 0.0117180        total: 23.9s    remaining: 1m 5s
268:    learn: 0.0116024        total: 24s      remaining: 1m 5s
269:    learn: 0.0114919        total: 24.1s    remaining: 1m 5s
270:    learn: 0.0113760        total: 24.2s    remaining: 1m 4s
271:    learn: 0.0112969        total: 24.2s    remaining: 1m 4s
272:    learn: 0.0111766        total: 24.3s    remaining: 1m 4s
273:    learn: 0.0110542        total: 24.4s    remaining: 1m 4s
274:    learn: 0.0109691        total: 24.5s    remaining: 1m 4s
275:    learn: 0.0108587        total: 24.6s    remaining: 1m 4s
276:    learn: 0.0107951        total: 24.7s    remaining: 1m 4s
277:    learn: 0.0107141        total: 24.8s    remaining: 1m 4s
278:    learn: 0.0106076        total: 24.9s    remaining: 1m 4s
279:    learn: 0.0105140        total: 24.9s    remaining: 1m 4s
280:    learn: 0.0104459        total: 25s      remaining: 1m 4s
281:    learn: 0.0103318        total: 25.1s    remaining: 1m 3s
282:    learn: 0.0102657        total: 25.2s    remaining: 1m 3s
283:    learn: 0.0101935        total: 25.3s    remaining: 1m 3s
284:    learn: 0.0100845        total: 25.4s    remaining: 1m 3s
285:    learn: 0.0100208        total: 25.5s    remaining: 1m 3s
286:    learn: 0.0099468        total: 25.6s    remaining: 1m 3s
287:    learn: 0.0099009        total: 25.6s    remaining: 1m 3s
288:    learn: 0.0098203        total: 25.7s    remaining: 1m 3s
289:    learn: 0.0097411        total: 25.8s    remaining: 1m 3s
290:    learn: 0.0096824        total: 25.9s    remaining: 1m 3s
291:    learn: 0.0096010        total: 26s      remaining: 1m 3s
292:    learn: 0.0095411        total: 26.1s    remaining: 1m 2s
293:    learn: 0.0094769        total: 26.2s    remaining: 1m 2s
294:    learn: 0.0093758        total: 26.3s    remaining: 1m 2s
295:    learn: 0.0092808        total: 26.3s    remaining: 1m 2s
296:    learn: 0.0091979        total: 26.4s    remaining: 1m 2s
297:    learn: 0.0091332        total: 26.5s    remaining: 1m 2s
298:    learn: 0.0090595        total: 26.6s    remaining: 1m 2s
299:    learn: 0.0089933        total: 26.7s    remaining: 1m 2s
300:    learn: 0.0089115        total: 26.8s    remaining: 1m 2s
301:    learn: 0.0088526        total: 26.9s    remaining: 1m 2s
302:    learn: 0.0087827        total: 27s      remaining: 1m 1s
303:    learn: 0.0087154        total: 27s      remaining: 1m 1s
304:    learn: 0.0086428        total: 27.1s    remaining: 1m 1s
305:    learn: 0.0085690        total: 27.2s    remaining: 1m 1s
306:    learn: 0.0085240        total: 27.3s    remaining: 1m 1s
307:    learn: 0.0084336        total: 27.4s    remaining: 1m 1s
308:    learn: 0.0083501        total: 27.5s    remaining: 1m 1s
309:    learn: 0.0083053        total: 27.6s    remaining: 1m 1s
```

```
310:    learn: 0.0082431        total: 27.6s    remaining: 1m 1s
311:    learn: 0.0081777        total: 27.7s    remaining: 1m 1s
312:    learn: 0.0081113        total: 27.8s    remaining: 1m 1s
313:    learn: 0.0080561        total: 27.9s    remaining: 1m
314:    learn: 0.0079937        total: 28s      remaining: 1m
315:    learn: 0.0079436        total: 28.1s    remaining: 1m
316:    learn: 0.0078687        total: 28.2s    remaining: 1m
317:    learn: 0.0078144        total: 28.2s    remaining: 1m
318:    learn: 0.0077437        total: 28.3s    remaining: 1m
319:    learn: 0.0076899        total: 28.4s    remaining: 1m
320:    learn: 0.0076220        total: 28.5s    remaining: 1m
321:    learn: 0.0075612        total: 28.6s    remaining: 1m
322:    learn: 0.0075126        total: 28.7s    remaining: 1m
323:    learn: 0.0074490        total: 28.8s    remaining: 1m
324:    learn: 0.0074017        total: 28.9s    remaining: 60s
325:    learn: 0.0073371        total: 29s      remaining: 59.9s
326:    learn: 0.0072862        total: 29s      remaining: 59.8s
327:    learn: 0.0072299        total: 29.1s    remaining: 59.7s
328:    learn: 0.0071769        total: 29.2s    remaining: 59.6s
329:    learn: 0.0071243        total: 29.3s    remaining: 59.5s
330:    learn: 0.0070713        total: 29.4s    remaining: 59.4s
331:    learn: 0.0070226        total: 29.5s    remaining: 59.3s
332:    learn: 0.0069722        total: 29.6s    remaining: 59.2s
333:    learn: 0.0069129        total: 29.6s    remaining: 59.1s
334:    learn: 0.0068581        total: 29.7s    remaining: 59s
335:    learn: 0.0067986        total: 29.8s    remaining: 58.9s
336:    learn: 0.0067520        total: 29.9s    remaining: 58.9s
337:    learn: 0.0066913        total: 30s      remaining: 58.8s
338:    learn: 0.0066428        total: 30.1s    remaining: 58.7s
339:    learn: 0.0065916        total: 30.2s    remaining: 58.6s
340:    learn: 0.0065471        total: 30.3s    remaining: 58.5s
341:    learn: 0.0065062        total: 30.4s    remaining: 58.4s
342:    learn: 0.0064732        total: 30.4s    remaining: 58.3s
343:    learn: 0.0064287        total: 30.5s    remaining: 58.2s
344:    learn: 0.0063619        total: 30.6s    remaining: 58.1s
345:    learn: 0.0063193        total: 30.7s    remaining: 58s
346:    learn: 0.0062584        total: 30.8s    remaining: 58s
347:    learn: 0.0062042        total: 30.9s    remaining: 57.9s
348:    learn: 0.0061684        total: 31s      remaining: 57.8s
349:    learn: 0.0061318        total: 31.1s    remaining: 57.7s
350:    learn: 0.0060846        total: 31.1s    remaining: 57.6s
351:    learn: 0.0060520        total: 31.2s    remaining: 57.5s
352:    learn: 0.0060110        total: 31.3s    remaining: 57.4s
353:    learn: 0.0059766        total: 31.4s    remaining: 57.3s
354:    learn: 0.0059458        total: 31.5s    remaining: 57.2s
355:    learn: 0.0059055        total: 31.6s    remaining: 57.1s
356:    learn: 0.0058551        total: 31.7s    remaining: 57.1s
357:    learn: 0.0058078        total: 31.8s    remaining: 57s
358:    learn: 0.0057692        total: 31.8s    remaining: 56.9s
359:    learn: 0.0057434        total: 31.9s    remaining: 56.8s
360:    learn: 0.0057007        total: 32s      remaining: 56.7s
361:    learn: 0.0056671        total: 32.1s    remaining: 56.6s
362:    learn: 0.0056257        total: 32.2s    remaining: 56.5s
363:    learn: 0.0056042        total: 32.3s    remaining: 56.4s
364:    learn: 0.0055734        total: 32.4s    remaining: 56.3s
365:    learn: 0.0055437        total: 32.5s    remaining: 56.2s
366:    learn: 0.0055182        total: 32.6s    remaining: 56.2s
367:    learn: 0.0054800        total: 32.6s    remaining: 56.1s
368:    learn: 0.0054400        total: 32.7s    remaining: 56s
369:    learn: 0.0054017        total: 32.8s    remaining: 55.9s
370:    learn: 0.0053661        total: 32.9s    remaining: 55.8s
371:    learn: 0.0053295        total: 33s      remaining: 55.7s
372:    learn: 0.0052902        total: 33.1s    remaining: 55.6s
373:    learn: 0.0052587        total: 33.2s    remaining: 55.5s
374:    learn: 0.0052325        total: 33.3s    remaining: 55.4s
375:    learn: 0.0051985        total: 33.3s    remaining: 55.3s
376:    learn: 0.0051731        total: 33.4s    remaining: 55.2s
377:    learn: 0.0051469        total: 33.5s    remaining: 55.2s
378:    learn: 0.0051166        total: 33.6s    remaining: 55.1s
```

```
379:    learn: 0.0050774        total: 33.7s    remaining: 55s
380:    learn: 0.0050475        total: 33.8s    remaining: 54.9s
381:    learn: 0.0050173        total: 33.9s    remaining: 54.8s
382:    learn: 0.0049820        total: 33.9s    remaining: 54.7s
383:    learn: 0.0049483        total: 34s      remaining: 54.6s
384:    learn: 0.0049219        total: 34.1s    remaining: 54.5s
385:    learn: 0.0048934        total: 34.2s    remaining: 54.4s
386:    learn: 0.0048664        total: 34.3s    remaining: 54.3s
387:    learn: 0.0048432        total: 34.4s    remaining: 54.2s
388:    learn: 0.0048110        total: 34.5s    remaining: 54.1s
389:    learn: 0.0047692        total: 34.6s    remaining: 54.1s
390:    learn: 0.0047415        total: 34.7s    remaining: 54s
391:    learn: 0.0047179        total: 34.7s    remaining: 53.9s
392:    learn: 0.0046861        total: 34.8s    remaining: 53.8s
393:    learn: 0.0046629        total: 34.9s    remaining: 53.7s
394:    learn: 0.0046398        total: 35s      remaining: 53.6s
395:    learn: 0.0046106        total: 35.1s    remaining: 53.5s
396:    learn: 0.0045838        total: 35.2s    remaining: 53.4s
397:    learn: 0.0045660        total: 35.3s    remaining: 53.3s
398:    learn: 0.0045284        total: 35.3s    remaining: 53.2s
399:    learn: 0.0044987        total: 35.4s    remaining: 53.1s
400:    learn: 0.0044810        total: 35.5s    remaining: 53.1s
401:    learn: 0.0044557        total: 35.6s    remaining: 53s
402:    learn: 0.0044321        total: 35.7s    remaining: 52.9s
403:    learn: 0.0044162        total: 35.8s    remaining: 52.8s
404:    learn: 0.0043871        total: 35.9s    remaining: 52.7s
405:    learn: 0.0043629        total: 36s      remaining: 52.6s
406:    learn: 0.0043379        total: 36.1s    remaining: 52.5s
407:    learn: 0.0043176        total: 36.1s    remaining: 52.4s
408:    learn: 0.0042963        total: 36.2s    remaining: 52.3s
409:    learn: 0.0042759        total: 36.3s    remaining: 52.3s
410:    learn: 0.0042455        total: 36.4s    remaining: 52.2s
411:    learn: 0.0042181        total: 36.5s    remaining: 52.1s
412:    learn: 0.0041915        total: 36.6s    remaining: 52s
413:    learn: 0.0041728        total: 36.7s    remaining: 51.9s
414:    learn: 0.0041498        total: 36.8s    remaining: 51.8s
415:    learn: 0.0041261        total: 36.9s    remaining: 51.7s
416:    learn: 0.0041006        total: 36.9s    remaining: 51.7s
417:    learn: 0.0040818        total: 37s      remaining: 51.6s
418:    learn: 0.0040665        total: 37.1s    remaining: 51.5s
419:    learn: 0.0040463        total: 37.2s    remaining: 51.4s
420:    learn: 0.0040237        total: 37.3s    remaining: 51.3s
421:    learn: 0.0040004        total: 37.4s    remaining: 51.2s
422:    learn: 0.0039827        total: 37.5s    remaining: 51.1s
423:    learn: 0.0039633        total: 37.6s    remaining: 51s
424:    learn: 0.0039490        total: 37.7s    remaining: 50.9s
425:    learn: 0.0039277        total: 37.7s    remaining: 50.9s
426:    learn: 0.0039049        total: 37.8s    remaining: 50.8s
427:    learn: 0.0038900        total: 37.9s    remaining: 50.7s
428:    learn: 0.0038679        total: 38s      remaining: 50.6s
429:    learn: 0.0038436        total: 38.1s    remaining: 50.5s
430:    learn: 0.0038197        total: 38.2s    remaining: 50.4s
431:    learn: 0.0038026        total: 38.3s    remaining: 50.3s
432:    learn: 0.0037834        total: 38.4s    remaining: 50.2s
433:    learn: 0.0037589        total: 38.5s    remaining: 50.1s
434:    learn: 0.0037412        total: 38.5s    remaining: 50.1s
435:    learn: 0.0037260        total: 38.6s    remaining: 50s
436:    learn: 0.0037066        total: 38.7s    remaining: 49.9s
437:    learn: 0.0036928        total: 38.8s    remaining: 49.8s
438:    learn: 0.0036778        total: 38.9s    remaining: 49.7s
439:    learn: 0.0036778        total: 39s      remaining: 49.6s
440:    learn: 0.0036640        total: 39.1s    remaining: 49.5s
441:    learn: 0.0036467        total: 39.1s    remaining: 49.4s
442:    learn: 0.0036291        total: 39.2s    remaining: 49.3s
443:    learn: 0.0036173        total: 39.3s    remaining: 49.2s
444:    learn: 0.0035991        total: 39.4s    remaining: 49.1s
445:    learn: 0.0035783        total: 39.5s    remaining: 49.1s
446:    learn: 0.0035645        total: 39.6s    remaining: 49s
447:    learn: 0.0035509        total: 39.7s    remaining: 48.9s
```

```
448:    learn: 0.0035315        total: 39.8s    remaining: 48.8s
449:    learn: 0.0035133        total: 39.8s    remaining: 48.7s
450:    learn: 0.0034970        total: 39.9s    remaining: 48.6s
451:    learn: 0.0034756        total: 40s      remaining: 48.5s
452:    learn: 0.0034576        total: 40.1s    remaining: 48.4s
453:    learn: 0.0034400        total: 40.2s    remaining: 48.3s
454:    learn: 0.0034241        total: 40.3s    remaining: 48.2s
455:    learn: 0.0034087        total: 40.4s    remaining: 48.1s
456:    learn: 0.0033975        total: 40.4s    remaining: 48s
457:    learn: 0.0033828        total: 40.5s    remaining: 48s
458:    learn: 0.0033644        total: 40.6s    remaining: 47.9s
459:    learn: 0.0033546        total: 40.7s    remaining: 47.8s
460:    learn: 0.0033393        total: 40.8s    remaining: 47.7s
461:    learn: 0.0033210        total: 40.9s    remaining: 47.6s
462:    learn: 0.0033079        total: 41s      remaining: 47.5s
463:    learn: 0.0032926        total: 41.1s    remaining: 47.5s
464:    learn: 0.0032823        total: 41.2s    remaining: 47.4s
465:    learn: 0.0032687        total: 41.3s    remaining: 47.3s
466:    learn: 0.0032557        total: 41.4s    remaining: 47.2s
467:    learn: 0.0032557        total: 41.4s    remaining: 47.1s
468:    learn: 0.0032440        total: 41.5s    remaining: 47s
469:    learn: 0.0032246        total: 41.6s    remaining: 46.9s
470:    learn: 0.0032138        total: 41.7s    remaining: 46.9s
471:    learn: 0.0032007        total: 41.8s    remaining: 46.8s
472:    learn: 0.0031835        total: 41.9s    remaining: 46.7s
473:    learn: 0.0031671        total: 42s      remaining: 46.6s
474:    learn: 0.0031462        total: 42.1s    remaining: 46.5s
475:    learn: 0.0031238        total: 42.2s    remaining: 46.4s
476:    learn: 0.0031115        total: 42.3s    remaining: 46.4s
477:    learn: 0.0030972        total: 42.4s    remaining: 46.3s
478:    learn: 0.0030836        total: 42.5s    remaining: 46.2s
479:    learn: 0.0030724        total: 42.6s    remaining: 46.1s
480:    learn: 0.0030546        total: 42.7s    remaining: 46s
481:    learn: 0.0030416        total: 42.8s    remaining: 46s
482:    learn: 0.0030318        total: 42.8s    remaining: 45.9s
483:    learn: 0.0030223        total: 42.9s    remaining: 45.8s
484:    learn: 0.0030094        total: 43s      remaining: 45.7s
485:    learn: 0.0029972        total: 43.1s    remaining: 45.6s
486:    learn: 0.0029972        total: 43.2s    remaining: 45.5s
487:    learn: 0.0029835        total: 43.3s    remaining: 45.4s
488:    learn: 0.0029705        total: 43.4s    remaining: 45.3s
489:    learn: 0.0029558        total: 43.5s    remaining: 45.2s
490:    learn: 0.0029464        total: 43.5s    remaining: 45.1s
491:    learn: 0.0029338        total: 43.6s    remaining: 45s
492:    learn: 0.0029182        total: 43.7s    remaining: 45s
493:    learn: 0.0029056        total: 43.8s    remaining: 44.9s
494:    learn: 0.0028912        total: 43.9s    remaining: 44.8s
495:    learn: 0.0028813        total: 44s      remaining: 44.7s
496:    learn: 0.0028691        total: 44.1s    remaining: 44.6s
497:    learn: 0.0028691        total: 44.2s    remaining: 44.5s
498:    learn: 0.0028691        total: 44.3s    remaining: 44.4s
499:    learn: 0.0028566        total: 44.3s    remaining: 44.3s
500:    learn: 0.0028419        total: 44.4s    remaining: 44.2s
501:    learn: 0.0028320        total: 44.5s    remaining: 44.2s
502:    learn: 0.0028150        total: 44.6s    remaining: 44.1s
503:    learn: 0.0028035        total: 44.7s    remaining: 44s
504:    learn: 0.0028035        total: 44.8s    remaining: 43.9s
505:    learn: 0.0027896        total: 44.9s    remaining: 43.8s
506:    learn: 0.0027718        total: 44.9s    remaining: 43.7s
507:    learn: 0.0027596        total: 45s      remaining: 43.6s
508:    learn: 0.0027489        total: 45.1s    remaining: 43.5s
509:    learn: 0.0027367        total: 45.2s    remaining: 43.4s
510:    learn: 0.0027231        total: 45.3s    remaining: 43.3s
511:    learn: 0.0027195        total: 45.4s    remaining: 43.3s
512:    learn: 0.0027070        total: 45.5s    remaining: 43.2s
513:    learn: 0.0026957        total: 45.6s    remaining: 43.1s
514:    learn: 0.0026957        total: 45.6s    remaining: 43s
515:    learn: 0.0026865        total: 45.7s    remaining: 42.9s
516:    learn: 0.0026762        total: 45.8s    remaining: 42.8s
```

```
517:    learn: 0.0026762        total: 45.9s    remaining: 42.7s
518:    learn: 0.0026606        total: 46s      remaining: 42.6s
519:    learn: 0.0026502        total: 46.1s    remaining: 42.5s
520:    learn: 0.0026383        total: 46.2s    remaining: 42.4s
521:    learn: 0.0026267        total: 46.2s    remaining: 42.3s
522:    learn: 0.0026267        total: 46.3s    remaining: 42.3s
523:    learn: 0.0026267        total: 46.4s    remaining: 42.2s
524:    learn: 0.0026171        total: 46.5s    remaining: 42.1s
525:    learn: 0.0026059        total: 46.6s    remaining: 42s
526:    learn: 0.0026059        total: 46.7s    remaining: 41.9s
527:    learn: 0.0025987        total: 46.8s    remaining: 41.8s
528:    learn: 0.0025987        total: 46.8s    remaining: 41.7s
529:    learn: 0.0025987        total: 46.9s    remaining: 41.6s
530:    learn: 0.0025987        total: 47s      remaining: 41.5s
531:    learn: 0.0025987        total: 47.1s    remaining: 41.4s
532:    learn: 0.0025986        total: 47.2s    remaining: 41.3s
533:    learn: 0.0025874        total: 47.3s    remaining: 41.3s
534:    learn: 0.0025785        total: 47.4s    remaining: 41.2s
535:    learn: 0.0025694        total: 47.4s    remaining: 41.1s
536:    learn: 0.0025602        total: 47.5s    remaining: 41s
537:    learn: 0.0025462        total: 47.6s    remaining: 40.9s
538:    learn: 0.0025461        total: 47.7s    remaining: 40.8s
539:    learn: 0.0025357        total: 47.8s    remaining: 40.7s
540:    learn: 0.0025357        total: 47.9s    remaining: 40.6s
541:    learn: 0.0025287        total: 48s      remaining: 40.5s
542:    learn: 0.0025199        total: 48s      remaining: 40.4s
543:    learn: 0.0025125        total: 48.1s    remaining: 40.3s
544:    learn: 0.0025125        total: 48.2s    remaining: 40.3s
545:    learn: 0.0025022        total: 48.3s    remaining: 40.2s
546:    learn: 0.0025022        total: 48.4s    remaining: 40.1s
547:    learn: 0.0024928        total: 48.5s    remaining: 40s
548:    learn: 0.0024848        total: 48.6s    remaining: 39.9s
549:    learn: 0.0024776        total: 48.7s    remaining: 39.8s
550:    learn: 0.0024683        total: 48.7s    remaining: 39.7s
551:    learn: 0.0024591        total: 48.8s    remaining: 39.6s
552:    learn: 0.0024591        total: 48.9s    remaining: 39.5s
553:    learn: 0.0024480        total: 49s      remaining: 39.4s
554:    learn: 0.0024449        total: 49.1s    remaining: 39.4s
555:    learn: 0.0024449        total: 49.2s    remaining: 39.3s
556:    learn: 0.0024359        total: 49.3s    remaining: 39.2s
557:    learn: 0.0024267        total: 49.4s    remaining: 39.1s
558:    learn: 0.0024267        total: 49.4s    remaining: 39s
559:    learn: 0.0024166        total: 49.5s    remaining: 38.9s
560:    learn: 0.0024092        total: 49.6s    remaining: 38.8s
561:    learn: 0.0023987        total: 49.7s    remaining: 38.7s
562:    learn: 0.0023987        total: 49.8s    remaining: 38.6s
563:    learn: 0.0023987        total: 49.9s    remaining: 38.5s
564:    learn: 0.0023986        total: 50s      remaining: 38.5s
565:    learn: 0.0023986        total: 50s      remaining: 38.4s
566:    learn: 0.0023894        total: 50.1s    remaining: 38.3s
567:    learn: 0.0023894        total: 50.2s    remaining: 38.2s
568:    learn: 0.0023893        total: 50.3s    remaining: 38.1s
569:    learn: 0.0023893        total: 50.4s    remaining: 38s
570:    learn: 0.0023818        total: 50.5s    remaining: 37.9s
571:    learn: 0.0023739        total: 50.6s    remaining: 37.8s
572:    learn: 0.0023739        total: 50.7s    remaining: 37.7s
573:    learn: 0.0023738        total: 50.7s    remaining: 37.7s
574:    learn: 0.0023649        total: 50.8s    remaining: 37.6s
575:    learn: 0.0023571        total: 50.9s    remaining: 37.5s
576:    learn: 0.0023470        total: 51s      remaining: 37.4s
577:    learn: 0.0023470        total: 51.1s    remaining: 37.3s
578:    learn: 0.0023470        total: 51.2s    remaining: 37.2s
579:    learn: 0.0023470        total: 51.3s    remaining: 37.1s
580:    learn: 0.0023470        total: 51.4s    remaining: 37s
581:    learn: 0.0023366        total: 51.4s    remaining: 36.9s
582:    learn: 0.0023366        total: 51.5s    remaining: 36.9s
583:    learn: 0.0023265        total: 51.6s    remaining: 36.8s
584:    learn: 0.0023265        total: 51.7s    remaining: 36.7s
585:    learn: 0.0023265        total: 51.8s    remaining: 36.6s
```

```
586:    learn: 0.0023265        total: 51.9s    remaining: 36.5s
587:    learn: 0.0023171        total: 52s      remaining: 36.4s
588:    learn: 0.0023084        total: 52s      remaining: 36.3s
589:    learn: 0.0023084        total: 52.1s    remaining: 36.2s
590:    learn: 0.0023002        total: 52.2s    remaining: 36.1s
591:    learn: 0.0023001        total: 52.3s    remaining: 36s
592:    learn: 0.0023001        total: 52.4s    remaining: 36s
593:    learn: 0.0023001        total: 52.5s    remaining: 35.9s
594:    learn: 0.0023001        total: 52.6s    remaining: 35.8s
595:    learn: 0.0022913        total: 52.7s    remaining: 35.7s
596:    learn: 0.0022913        total: 52.7s    remaining: 35.6s
597:    learn: 0.0022840        total: 52.8s    remaining: 35.5s
598:    learn: 0.0022767        total: 52.9s    remaining: 35.4s
599:    learn: 0.0022682        total: 53s      remaining: 35.3s
600:    learn: 0.0022682        total: 53.1s    remaining: 35.2s
601:    learn: 0.0022569        total: 53.2s    remaining: 35.2s
602:    learn: 0.0022569        total: 53.3s    remaining: 35.1s
603:    learn: 0.0022478        total: 53.3s    remaining: 35s
604:    learn: 0.0022435        total: 53.4s    remaining: 34.9s
605:    learn: 0.0022346        total: 53.5s    remaining: 34.8s
606:    learn: 0.0022267        total: 53.6s    remaining: 34.7s
607:    learn: 0.0022267        total: 53.7s    remaining: 34.6s
608:    learn: 0.0022267        total: 53.8s    remaining: 34.5s
609:    learn: 0.0022266        total: 53.9s    remaining: 34.4s
610:    learn: 0.0022266        total: 53.9s    remaining: 34.3s
611:    learn: 0.0022199        total: 54s      remaining: 34.3s
612:    learn: 0.0022199        total: 54.1s    remaining: 34.2s
613:    learn: 0.0022199        total: 54.2s    remaining: 34.1s
614:    learn: 0.0022132        total: 54.3s    remaining: 34s
615:    learn: 0.0022089        total: 54.4s    remaining: 33.9s
616:    learn: 0.0022023        total: 54.5s    remaining: 33.8s
617:    learn: 0.0021969        total: 54.6s    remaining: 33.7s
618:    learn: 0.0021969        total: 54.6s    remaining: 33.6s
619:    learn: 0.0021969        total: 54.7s    remaining: 33.5s
620:    learn: 0.0021969        total: 54.8s    remaining: 33.4s
621:    learn: 0.0021969        total: 54.9s    remaining: 33.4s
622:    learn: 0.0021968        total: 55s      remaining: 33.3s
623:    learn: 0.0021968        total: 55.1s    remaining: 33.2s
624:    learn: 0.0021968        total: 55.2s    remaining: 33.1s
625:    learn: 0.0021864        total: 55.2s    remaining: 33s
626:    learn: 0.0021863        total: 55.3s    remaining: 32.9s
627:    learn: 0.0021798        total: 55.4s    remaining: 32.8s
628:    learn: 0.0021798        total: 55.5s    remaining: 32.7s
629:    learn: 0.0021798        total: 55.6s    remaining: 32.6s
630:    learn: 0.0021797        total: 55.7s    remaining: 32.6s
631:    learn: 0.0021715        total: 55.8s    remaining: 32.5s
632:    learn: 0.0021647        total: 55.9s    remaining: 32.4s
633:    learn: 0.0021647        total: 55.9s    remaining: 32.3s
634:    learn: 0.0021553        total: 56s      remaining: 32.2s
635:    learn: 0.0021494        total: 56.1s    remaining: 32.1s
636:    learn: 0.0021494        total: 56.2s    remaining: 32s
637:    learn: 0.0021494        total: 56.3s    remaining: 31.9s
638:    learn: 0.0021494        total: 56.4s    remaining: 31.8s
639:    learn: 0.0021494        total: 56.5s    remaining: 31.8s
640:    learn: 0.0021493        total: 56.5s    remaining: 31.7s
641:    learn: 0.0021404        total: 56.6s    remaining: 31.6s
642:    learn: 0.0021320        total: 56.7s    remaining: 31.5s
643:    learn: 0.0021320        total: 56.8s    remaining: 31.4s
644:    learn: 0.0021320        total: 56.9s    remaining: 31.3s
645:    learn: 0.0021320        total: 57s      remaining: 31.2s
646:    learn: 0.0021320        total: 57.1s    remaining: 31.1s
647:    learn: 0.0021319        total: 57.2s    remaining: 31s
648:    learn: 0.0021319        total: 57.2s    remaining: 31s
649:    learn: 0.0021235        total: 57.3s    remaining: 30.9s
650:    learn: 0.0021182        total: 57.4s    remaining: 30.8s
651:    learn: 0.0021182        total: 57.5s    remaining: 30.7s
652:    learn: 0.0021182        total: 57.6s    remaining: 30.6s
653:    learn: 0.0021182        total: 57.7s    remaining: 30.5s
654:    learn: 0.0021182        total: 57.8s    remaining: 30.4s
```

```
655:    learn: 0.0021182        total: 57.9s    remaining: 30.3s
656:    learn: 0.0021182        total: 57.9s    remaining: 30.2s
657:    learn: 0.0021182        total: 58s      remaining: 30.2s
658:    learn: 0.0021182        total: 58.1s    remaining: 30.1s
659:    learn: 0.0021124        total: 58.2s    remaining: 30s
660:    learn: 0.0021063        total: 58.3s    remaining: 29.9s
661:    learn: 0.0021063        total: 58.4s    remaining: 29.8s
662:    learn: 0.0021062        total: 58.4s    remaining: 29.7s
663:    learn: 0.0021062        total: 58.5s    remaining: 29.6s
664:    learn: 0.0021036        total: 58.6s    remaining: 29.5s
665:    learn: 0.0021036        total: 58.7s    remaining: 29.4s
666:    learn: 0.0020938        total: 58.8s    remaining: 29.3s
667:    learn: 0.0020938        total: 58.9s    remaining: 29.3s
668:    learn: 0.0020938        total: 59s      remaining: 29.2s
669:    learn: 0.0020852        total: 59s      remaining: 29.1s
670:    learn: 0.0020852        total: 59.1s    remaining: 29s
671:    learn: 0.0020782        total: 59.2s    remaining: 28.9s
672:    learn: 0.0020782        total: 59.3s    remaining: 28.8s
673:    learn: 0.0020782        total: 59.4s    remaining: 28.7s
674:    learn: 0.0020782        total: 59.5s    remaining: 28.6s
675:    learn: 0.0020750        total: 59.6s    remaining: 28.5s
676:    learn: 0.0020750        total: 59.6s    remaining: 28.5s
677:    learn: 0.0020750        total: 59.7s    remaining: 28.4s
678:    learn: 0.0020750        total: 59.8s    remaining: 28.3s
679:    learn: 0.0020750        total: 59.9s    remaining: 28.2s
680:    learn: 0.0020750        total: 60s      remaining: 28.1s
681:    learn: 0.0020750        total: 1m       remaining: 28s
682:    learn: 0.0020750        total: 1m       remaining: 27.9s
683:    learn: 0.0020750        total: 1m       remaining: 27.8s
684:    learn: 0.0020750        total: 1m       remaining: 27.7s
685:    learn: 0.0020750        total: 1m       remaining: 27.7s
686:    learn: 0.0020694        total: 1m       remaining: 27.6s
687:    learn: 0.0020694        total: 1m       remaining: 27.5s
688:    learn: 0.0020694        total: 1m       remaining: 27.4s
689:    learn: 0.0020693        total: 1m       remaining: 27.3s
690:    learn: 0.0020693        total: 1m       remaining: 27.2s
691:    learn: 0.0020693        total: 1m       remaining: 27.1s
692:    learn: 0.0020693        total: 1m 1s    remaining: 27s
693:    learn: 0.0020693        total: 1m 1s    remaining: 26.9s
694:    learn: 0.0020693        total: 1m 1s    remaining: 26.9s
695:    learn: 0.0020693        total: 1m 1s    remaining: 26.8s
696:    learn: 0.0020692        total: 1m 1s    remaining: 26.7s
697:    learn: 0.0020692        total: 1m 1s    remaining: 26.6s
698:    learn: 0.0020692        total: 1m 1s    remaining: 26.5s
699:    learn: 0.0020692        total: 1m 1s    remaining: 26.4s
700:    learn: 0.0020685        total: 1m 1s    remaining: 26.3s
701:    learn: 0.0020685        total: 1m 1s    remaining: 26.2s
702:    learn: 0.0020684        total: 1m 1s    remaining: 26.2s
703:    learn: 0.0020684        total: 1m 1s    remaining: 26.1s
704:    learn: 0.0020684        total: 1m 2s    remaining: 26s
705:    learn: 0.0020619        total: 1m 2s    remaining: 25.9s
706:    learn: 0.0020618        total: 1m 2s    remaining: 25.8s
707:    learn: 0.0020618        total: 1m 2s    remaining: 25.7s
708:    learn: 0.0020618        total: 1m 2s    remaining: 25.6s
709:    learn: 0.0020618        total: 1m 2s    remaining: 25.5s
710:    learn: 0.0020618        total: 1m 2s    remaining: 25.4s
711:    learn: 0.0020618        total: 1m 2s    remaining: 25.4s
712:    learn: 0.0020618        total: 1m 2s    remaining: 25.3s
713:    learn: 0.0020618        total: 1m 2s    remaining: 25.2s
714:    learn: 0.0020618        total: 1m 2s    remaining: 25.1s
715:    learn: 0.0020618        total: 1m 3s    remaining: 25s
716:    learn: 0.0020618        total: 1m 3s    remaining: 24.9s
717:    learn: 0.0020618        total: 1m 3s    remaining: 24.8s
718:    learn: 0.0020618        total: 1m 3s    remaining: 24.7s
719:    learn: 0.0020550        total: 1m 3s    remaining: 24.6s
720:    learn: 0.0020550        total: 1m 3s    remaining: 24.6s
721:    learn: 0.0020550        total: 1m 3s    remaining: 24.5s
722:    learn: 0.0020549        total: 1m 3s    remaining: 24.4s
723:    learn: 0.0020549        total: 1m 3s    remaining: 24.3s
```

```
724:    learn: 0.0020549        total: 1m 3s    remaining: 24.2s
725:    learn: 0.0020455        total: 1m 3s    remaining: 24.1s
726:    learn: 0.0020455        total: 1m 3s    remaining: 24s
727:    learn: 0.0020455        total: 1m 4s    remaining: 23.9s
728:    learn: 0.0020455        total: 1m 4s    remaining: 23.8s
729:    learn: 0.0020382        total: 1m 4s    remaining: 23.8s
730:    learn: 0.0020382        total: 1m 4s    remaining: 23.7s
731:    learn: 0.0020382        total: 1m 4s    remaining: 23.6s
732:    learn: 0.0020382        total: 1m 4s    remaining: 23.5s
733:    learn: 0.0020382        total: 1m 4s    remaining: 23.4s
734:    learn: 0.0020382        total: 1m 4s    remaining: 23.3s
735:    learn: 0.0020382        total: 1m 4s    remaining: 23.2s
736:    learn: 0.0020382        total: 1m 4s    remaining: 23.1s
737:    learn: 0.0020382        total: 1m 4s    remaining: 23.1s
738:    learn: 0.0020382        total: 1m 5s    remaining: 23s
739:    learn: 0.0020382        total: 1m 5s    remaining: 22.9s
740:    learn: 0.0020382        total: 1m 5s    remaining: 22.8s
741:    learn: 0.0020381        total: 1m 5s    remaining: 22.7s
742:    learn: 0.0020381        total: 1m 5s    remaining: 22.6s
743:    learn: 0.0020381        total: 1m 5s    remaining: 22.5s
744:    learn: 0.0020381        total: 1m 5s    remaining: 22.4s
745:    learn: 0.0020314        total: 1m 5s    remaining: 22.3s
746:    learn: 0.0020314        total: 1m 5s    remaining: 22.3s
747:    learn: 0.0020314        total: 1m 5s    remaining: 22.2s
748:    learn: 0.0020314        total: 1m 5s    remaining: 22.1s
749:    learn: 0.0020314        total: 1m 5s    remaining: 22s
750:    learn: 0.0020314        total: 1m 6s    remaining: 21.9s
751:    learn: 0.0020314        total: 1m 6s    remaining: 21.8s
752:    learn: 0.0020314        total: 1m 6s    remaining: 21.7s
753:    learn: 0.0020313        total: 1m 6s    remaining: 21.6s
754:    learn: 0.0020313        total: 1m 6s    remaining: 21.5s
755:    learn: 0.0020313        total: 1m 6s    remaining: 21.5s
756:    learn: 0.0020313        total: 1m 6s    remaining: 21.4s
757:    learn: 0.0020234        total: 1m 6s    remaining: 21.3s
758:    learn: 0.0020168        total: 1m 6s    remaining: 21.2s
759:    learn: 0.0020168        total: 1m 6s    remaining: 21.1s
760:    learn: 0.0020168        total: 1m 6s    remaining: 21s
761:    learn: 0.0020086        total: 1m 7s    remaining: 20.9s
762:    learn: 0.0020086        total: 1m 7s    remaining: 20.8s
763:    learn: 0.0020085        total: 1m 7s    remaining: 20.7s
764:    learn: 0.0020085        total: 1m 7s    remaining: 20.7s
765:    learn: 0.0020085        total: 1m 7s    remaining: 20.6s
766:    learn: 0.0020085        total: 1m 7s    remaining: 20.5s
767:    learn: 0.0020085        total: 1m 7s    remaining: 20.4s
768:    learn: 0.0020033        total: 1m 7s    remaining: 20.3s
769:    learn: 0.0020033        total: 1m 7s    remaining: 20.2s
770:    learn: 0.0020033        total: 1m 7s    remaining: 20.1s
771:    learn: 0.0020033        total: 1m 7s    remaining: 20s
772:    learn: 0.0020033        total: 1m 7s    remaining: 20s
773:    learn: 0.0020033        total: 1m 8s    remaining: 19.9s
774:    learn: 0.0020033        total: 1m 8s    remaining: 19.8s
775:    learn: 0.0020032        total: 1m 8s    remaining: 19.7s
776:    learn: 0.0020032        total: 1m 8s    remaining: 19.6s
777:    learn: 0.0020032        total: 1m 8s    remaining: 19.5s
778:    learn: 0.0020032        total: 1m 8s    remaining: 19.4s
779:    learn: 0.0019962        total: 1m 8s    remaining: 19.3s
780:    learn: 0.0019962        total: 1m 8s    remaining: 19.2s
781:    learn: 0.0019962        total: 1m 8s    remaining: 19.2s
782:    learn: 0.0019962        total: 1m 8s    remaining: 19.1s
783:    learn: 0.0019962        total: 1m 8s    remaining: 19s
784:    learn: 0.0019962        total: 1m 8s    remaining: 18.9s
785:    learn: 0.0019962        total: 1m 9s    remaining: 18.8s
786:    learn: 0.0019962        total: 1m 9s    remaining: 18.7s
787:    learn: 0.0019962        total: 1m 9s    remaining: 18.6s
788:    learn: 0.0019962        total: 1m 9s    remaining: 18.5s
789:    learn: 0.0019886        total: 1m 9s    remaining: 18.5s
790:    learn: 0.0019886        total: 1m 9s    remaining: 18.4s
791:    learn: 0.0019886        total: 1m 9s    remaining: 18.3s
792:    learn: 0.0019820        total: 1m 9s    remaining: 18.2s
```

```
793:    learn: 0.0019820        total: 1m 9s    remaining: 18.1s
794:    learn: 0.0019819        total: 1m 9s    remaining: 18s
795:    learn: 0.0019819        total: 1m 9s    remaining: 17.9s
796:    learn: 0.0019751        total: 1m 10s   remaining: 17.8s
797:    learn: 0.0019751        total: 1m 10s   remaining: 17.8s
798:    learn: 0.0019751        total: 1m 10s   remaining: 17.7s
799:    learn: 0.0019750        total: 1m 10s   remaining: 17.6s
800:    learn: 0.0019750        total: 1m 10s   remaining: 17.5s
801:    learn: 0.0019678        total: 1m 10s   remaining: 17.4s
802:    learn: 0.0019678        total: 1m 10s   remaining: 17.3s
803:    learn: 0.0019639        total: 1m 10s   remaining: 17.2s
804:    learn: 0.0019639        total: 1m 10s   remaining: 17.1s
805:    learn: 0.0019639        total: 1m 10s   remaining: 17s
806:    learn: 0.0019639        total: 1m 10s   remaining: 17s
807:    learn: 0.0019639        total: 1m 10s   remaining: 16.9s
808:    learn: 0.0019638        total: 1m 11s   remaining: 16.8s
809:    learn: 0.0019638        total: 1m 11s   remaining: 16.7s
810:    learn: 0.0019638        total: 1m 11s   remaining: 16.6s
811:    learn: 0.0019638        total: 1m 11s   remaining: 16.5s
812:    learn: 0.0019568        total: 1m 11s   remaining: 16.4s
813:    learn: 0.0019497        total: 1m 11s   remaining: 16.3s
814:    learn: 0.0019497        total: 1m 11s   remaining: 16.3s
815:    learn: 0.0019423        total: 1m 11s   remaining: 16.2s
816:    learn: 0.0019423        total: 1m 11s   remaining: 16.1s
817:    learn: 0.0019423        total: 1m 11s   remaining: 16s
818:    learn: 0.0019423        total: 1m 11s   remaining: 15.9s
819:    learn: 0.0019423        total: 1m 12s   remaining: 15.8s
820:    learn: 0.0019423        total: 1m 12s   remaining: 15.7s
821:    learn: 0.0019392        total: 1m 12s   remaining: 15.6s
822:    learn: 0.0019392        total: 1m 12s   remaining: 15.6s
823:    learn: 0.0019329        total: 1m 12s   remaining: 15.5s
824:    learn: 0.0019329        total: 1m 12s   remaining: 15.4s
825:    learn: 0.0019329        total: 1m 12s   remaining: 15.3s
826:    learn: 0.0019329        total: 1m 12s   remaining: 15.2s
827:    learn: 0.0019328        total: 1m 12s   remaining: 15.1s
828:    learn: 0.0019328        total: 1m 12s   remaining: 15s
829:    learn: 0.0019328        total: 1m 12s   remaining: 14.9s
830:    learn: 0.0019328        total: 1m 13s   remaining: 14.9s
831:    learn: 0.0019328        total: 1m 13s   remaining: 14.8s
832:    learn: 0.0019328        total: 1m 13s   remaining: 14.7s
833:    learn: 0.0019328        total: 1m 13s   remaining: 14.6s
834:    learn: 0.0019277        total: 1m 13s   remaining: 14.5s
835:    learn: 0.0019277        total: 1m 13s   remaining: 14.4s
836:    learn: 0.0019277        total: 1m 13s   remaining: 14.3s
837:    learn: 0.0019277        total: 1m 13s   remaining: 14.2s
838:    learn: 0.0019277        total: 1m 13s   remaining: 14.1s
839:    learn: 0.0019276        total: 1m 13s   remaining: 14.1s
840:    learn: 0.0019276        total: 1m 13s   remaining: 14s
841:    learn: 0.0019275        total: 1m 14s   remaining: 13.9s
842:    learn: 0.0019275        total: 1m 14s   remaining: 13.8s
843:    learn: 0.0019275        total: 1m 14s   remaining: 13.7s
844:    learn: 0.0019275        total: 1m 14s   remaining: 13.6s
845:    learn: 0.0019275        total: 1m 14s   remaining: 13.5s
846:    learn: 0.0019275        total: 1m 14s   remaining: 13.4s
847:    learn: 0.0019275        total: 1m 14s   remaining: 13.4s
848:    learn: 0.0019275        total: 1m 14s   remaining: 13.3s
849:    learn: 0.0019260        total: 1m 14s   remaining: 13.2s
850:    learn: 0.0019259        total: 1m 14s   remaining: 13.1s
851:    learn: 0.0019250        total: 1m 14s   remaining: 13s
852:    learn: 0.0019250        total: 1m 14s   remaining: 12.9s
853:    learn: 0.0019250        total: 1m 15s   remaining: 12.8s
854:    learn: 0.0019185        total: 1m 15s   remaining: 12.7s
855:    learn: 0.0019185        total: 1m 15s   remaining: 12.7s
856:    learn: 0.0019185        total: 1m 15s   remaining: 12.6s
857:    learn: 0.0019185        total: 1m 15s   remaining: 12.5s
858:    learn: 0.0019185        total: 1m 15s   remaining: 12.4s
859:    learn: 0.0019139        total: 1m 15s   remaining: 12.3s
860:    learn: 0.0019139        total: 1m 15s   remaining: 12.2s
861:    learn: 0.0019139        total: 1m 15s   remaining: 12.1s
```

```
862:    learn: 0.0019085        total: 1m 15s   remaining: 12s
863:    learn: 0.0019085        total: 1m 15s   remaining: 12s
864:    learn: 0.0019085        total: 1m 16s   remaining: 11.9s
865:    learn: 0.0019085        total: 1m 16s   remaining: 11.8s
866:    learn: 0.0019085        total: 1m 16s   remaining: 11.7s
867:    learn: 0.0019085        total: 1m 16s   remaining: 11.6s
868:    learn: 0.0019085        total: 1m 16s   remaining: 11.5s
869:    learn: 0.0019085        total: 1m 16s   remaining: 11.4s
870:    learn: 0.0019085        total: 1m 16s   remaining: 11.3s
871:    learn: 0.0019084        total: 1m 16s   remaining: 11.2s
872:    learn: 0.0019034        total: 1m 16s   remaining: 11.2s
873:    learn: 0.0019034        total: 1m 16s   remaining: 11.1s
874:    learn: 0.0019034        total: 1m 16s   remaining: 11s
875:    learn: 0.0019034        total: 1m 16s   remaining: 10.9s
876:    learn: 0.0019034        total: 1m 17s   remaining: 10.8s
877:    learn: 0.0018966        total: 1m 17s   remaining: 10.7s
878:    learn: 0.0018966        total: 1m 17s   remaining: 10.6s
879:    learn: 0.0018966        total: 1m 17s   remaining: 10.5s
880:    learn: 0.0018966        total: 1m 17s   remaining: 10.5s
881:    learn: 0.0018966        total: 1m 17s   remaining: 10.4s
882:    learn: 0.0018937        total: 1m 17s   remaining: 10.3s
883:    learn: 0.0018937        total: 1m 17s   remaining: 10.2s
884:    learn: 0.0018872        total: 1m 17s   remaining: 10.1s
885:    learn: 0.0018872        total: 1m 17s   remaining: 10s
886:    learn: 0.0018872        total: 1m 17s   remaining: 9.93s
887:    learn: 0.0018872        total: 1m 18s   remaining: 9.84s
888:    learn: 0.0018872        total: 1m 18s   remaining: 9.75s
889:    learn: 0.0018871        total: 1m 18s   remaining: 9.66s
890:    learn: 0.0018871        total: 1m 18s   remaining: 9.58s
891:    learn: 0.0018871        total: 1m 18s   remaining: 9.49s
892:    learn: 0.0018871        total: 1m 18s   remaining: 9.4s
893:    learn: 0.0018871        total: 1m 18s   remaining: 9.31s
894:    learn: 0.0018871        total: 1m 18s   remaining: 9.23s
895:    learn: 0.0018871        total: 1m 18s   remaining: 9.14s
896:    learn: 0.0018871        total: 1m 18s   remaining: 9.05s
897:    learn: 0.0018871        total: 1m 18s   remaining: 8.96s
898:    learn: 0.0018871        total: 1m 19s   remaining: 8.88s
899:    learn: 0.0018871        total: 1m 19s   remaining: 8.79s
900:    learn: 0.0018780        total: 1m 19s   remaining: 8.7s
901:    learn: 0.0018780        total: 1m 19s   remaining: 8.61s
902:    learn: 0.0018780        total: 1m 19s   remaining: 8.53s
903:    learn: 0.0018719        total: 1m 19s   remaining: 8.44s
904:    learn: 0.0018719        total: 1m 19s   remaining: 8.35s
905:    learn: 0.0018718        total: 1m 19s   remaining: 8.26s
906:    learn: 0.0018718        total: 1m 19s   remaining: 8.18s
907:    learn: 0.0018718        total: 1m 19s   remaining: 8.09s
908:    learn: 0.0018718        total: 1m 19s   remaining: 8s
909:    learn: 0.0018718        total: 1m 20s   remaining: 7.91s
910:    learn: 0.0018718        total: 1m 20s   remaining: 7.83s
911:    learn: 0.0018718        total: 1m 20s   remaining: 7.74s
912:    learn: 0.0018718        total: 1m 20s   remaining: 7.65s
913:    learn: 0.0018652        total: 1m 20s   remaining: 7.56s
914:    learn: 0.0018652        total: 1m 20s   remaining: 7.48s
915:    learn: 0.0018568        total: 1m 20s   remaining: 7.39s
916:    learn: 0.0018568        total: 1m 20s   remaining: 7.3s
917:    learn: 0.0018568        total: 1m 20s   remaining: 7.21s
918:    learn: 0.0018568        total: 1m 20s   remaining: 7.13s
919:    learn: 0.0018568        total: 1m 20s   remaining: 7.04s
920:    learn: 0.0018568        total: 1m 21s   remaining: 6.95s
921:    learn: 0.0018568        total: 1m 21s   remaining: 6.86s
922:    learn: 0.0018568        total: 1m 21s   remaining: 6.77s
923:    learn: 0.0018568        total: 1m 21s   remaining: 6.69s
924:    learn: 0.0018568        total: 1m 21s   remaining: 6.6s
925:    learn: 0.0018568        total: 1m 21s   remaining: 6.51s
926:    learn: 0.0018568        total: 1m 21s   remaining: 6.42s
927:    learn: 0.0018567        total: 1m 21s   remaining: 6.33s
928:    learn: 0.0018567        total: 1m 21s   remaining: 6.25s
929:    learn: 0.0018567        total: 1m 21s   remaining: 6.16s
930:    learn: 0.0018566        total: 1m 21s   remaining: 6.07s
```

```
931:    learn: 0.0018508        total: 1m 21s   remaining: 5.98s
932:    learn: 0.0018508        total: 1m 22s   remaining: 5.89s
933:    learn: 0.0018508        total: 1m 22s   remaining: 5.81s
934:    learn: 0.0018508        total: 1m 22s   remaining: 5.72s
935:    learn: 0.0018507        total: 1m 22s   remaining: 5.63s
936:    learn: 0.0018507        total: 1m 22s   remaining: 5.54s
937:    learn: 0.0018507        total: 1m 22s   remaining: 5.45s
938:    learn: 0.0018507        total: 1m 22s   remaining: 5.37s
939:    learn: 0.0018507        total: 1m 22s   remaining: 5.28s
940:    learn: 0.0018507        total: 1m 22s   remaining: 5.19s
941:    learn: 0.0018507        total: 1m 22s   remaining: 5.1s
942:    learn: 0.0018507        total: 1m 22s   remaining: 5.01s
943:    learn: 0.0018507        total: 1m 23s   remaining: 4.93s
944:    learn: 0.0018507        total: 1m 23s   remaining: 4.84s
945:    learn: 0.0018507        total: 1m 23s   remaining: 4.75s
946:    learn: 0.0018507        total: 1m 23s   remaining: 4.66s
947:    learn: 0.0018507        total: 1m 23s   remaining: 4.57s
948:    learn: 0.0018506        total: 1m 23s   remaining: 4.49s
949:    learn: 0.0018506        total: 1m 23s   remaining: 4.4s
950:    learn: 0.0018506        total: 1m 23s   remaining: 4.31s
951:    learn: 0.0018506        total: 1m 23s   remaining: 4.22s
952:    learn: 0.0018506        total: 1m 23s   remaining: 4.13s
953:    learn: 0.0018435        total: 1m 23s   remaining: 4.04s
954:    learn: 0.0018435        total: 1m 23s   remaining: 3.96s
955:    learn: 0.0018435        total: 1m 24s   remaining: 3.87s
956:    learn: 0.0018435        total: 1m 24s   remaining: 3.78s
957:    learn: 0.0018435        total: 1m 24s   remaining: 3.69s
958:    learn: 0.0018434        total: 1m 24s   remaining: 3.6s
959:    learn: 0.0018420        total: 1m 24s   remaining: 3.52s
960:    learn: 0.0018420        total: 1m 24s   remaining: 3.43s
961:    learn: 0.0018419        total: 1m 24s   remaining: 3.34s
962:    learn: 0.0018419        total: 1m 24s   remaining: 3.25s
963:    learn: 0.0018419        total: 1m 24s   remaining: 3.17s
964:    learn: 0.0018419        total: 1m 24s   remaining: 3.08s
965:    learn: 0.0018419        total: 1m 24s   remaining: 2.99s
966:    learn: 0.0018419        total: 1m 25s   remaining: 2.9s
967:    learn: 0.0018419        total: 1m 25s   remaining: 2.81s
968:    learn: 0.0018419        total: 1m 25s   remaining: 2.73s
969:    learn: 0.0018419        total: 1m 25s   remaining: 2.64s
970:    learn: 0.0018419        total: 1m 25s   remaining: 2.55s
971:    learn: 0.0018419        total: 1m 25s   remaining: 2.46s
972:    learn: 0.0018419        total: 1m 25s   remaining: 2.37s
973:    learn: 0.0018419        total: 1m 25s   remaining: 2.29s
974:    learn: 0.0018419        total: 1m 25s   remaining: 2.2s
975:    learn: 0.0018419        total: 1m 25s   remaining: 2.11s
976:    learn: 0.0018419        total: 1m 25s   remaining: 2.02s
977:    learn: 0.0018419        total: 1m 25s   remaining: 1.93s
978:    learn: 0.0018419        total: 1m 26s   remaining: 1.85s
979:    learn: 0.0018419        total: 1m 26s   remaining: 1.76s
980:    learn: 0.0018419        total: 1m 26s   remaining: 1.67s
981:    learn: 0.0018419        total: 1m 26s   remaining: 1.58s
982:    learn: 0.0018406        total: 1m 26s   remaining: 1.49s
983:    learn: 0.0018406        total: 1m 26s   remaining: 1.41s
984:    learn: 0.0018406        total: 1m 26s   remaining: 1.32s
985:    learn: 0.0018406        total: 1m 26s   remaining: 1.23s
986:    learn: 0.0018406        total: 1m 26s   remaining: 1.14s
987:    learn: 0.0018406        total: 1m 26s   remaining: 1.05s
988:    learn: 0.0018406        total: 1m 26s   remaining: 967ms
989:    learn: 0.0018406        total: 1m 27s   remaining: 879ms
990:    learn: 0.0018406        total: 1m 27s   remaining: 791ms
991:    learn: 0.0018406        total: 1m 27s   remaining: 703ms
992:    learn: 0.0018405        total: 1m 27s   remaining: 615ms
993:    learn: 0.0018405        total: 1m 27s   remaining: 527ms
994:    learn: 0.0018405        total: 1m 27s   remaining: 440ms
995:    learn: 0.0018405        total: 1m 27s   remaining: 352ms
996:    learn: 0.0018405        total: 1m 27s   remaining: 264ms
997:    learn: 0.0018405        total: 1m 27s   remaining: 176ms
998:    learn: 0.0018405        total: 1m 27s   remaining: 87.9ms
999:    learn: 0.0018405        total: 1m 27s   remaining: 0us
```

Out[136]:

```
StackingCVClassifier(classifiers=[ExtraTreesClassifier(max_features=0.5,
                                                       n_estimators=200,
                                                       n_jobs=-1,
                                                       random_state=33),
                                  SVC(C=100.0, gamma=0.001, max_iter=10000,
                                      random_state=33),
                                  XGBClassifier(base_score=0.5,
                                                booster='gbtree',
                                                colsample_bylevel=1,
                                                colsample_bynode=1,
                                                colsample_bytree=0.6, gamma=0,
                                                gpu_id=-1,
                                                importance_type='gain',
                                                interaction_constraints='',
                                                lear...
                                                 min_split_gain=0.5,
                                                 n_estimators=400,
                                                 num_leaves=26,
                                                 objective='binary',
                                                 random_state=33,
                                                 scale_pos_weight=1,
                                                 subsample=0.8012317612884369,
                                                 subsample_for_bin=200,
                                                 subsample_freq=1),
                                  KNeighborsClassifier(n_jobs=-1, n_neighbors=1,
                                                       p=1)],
                      cv=5,
                      meta_classifier=LogisticRegression(C=0.1, max_iter=10000,
                                                         penalty='l1',
                                                         random_state=33,
                                                         solver='liblinear'),
                      n jobs=-1, random state=33)
```

12.3 Assess the performance

In [158]:

```
estimators.append(sclf)
print('Training accuracy: {:.2f}%'.format(
    sclf.score(X_important_train, y_train) * 100))
print('Test accuracy:     {:.2f}%'.format(
    sclf.score(X_important_test, y_test) * 100))
```

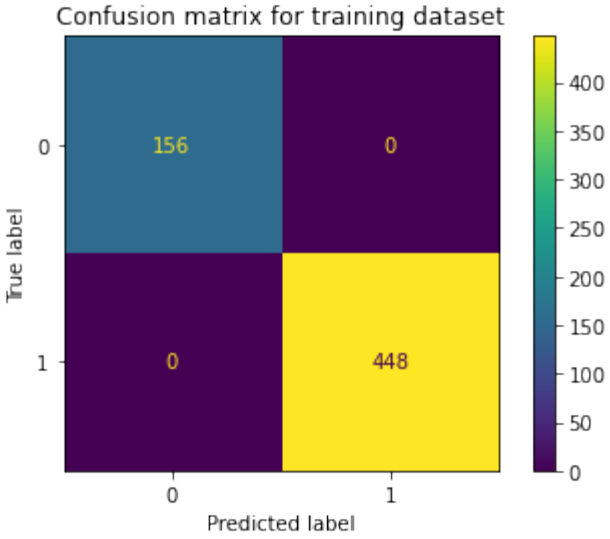
```
Training accuracy: 100.00%
Test accuracy:     97.37%
```

In [139]:

```
print('Performance on training data:\n')
print(
    classification_report(y_train,
                          sclf.predict(X_important_train),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(sclf, X_important_train, y_train, labels=[0, 1])
plt.title('Confusion matrix for training dataset')
plt.show()
```

```
Performance on training data:

              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000       156
           1     1.0000    1.0000    1.0000       448

    accuracy                         1.0000       604
   macro avg     1.0000    1.0000    1.0000       604
```

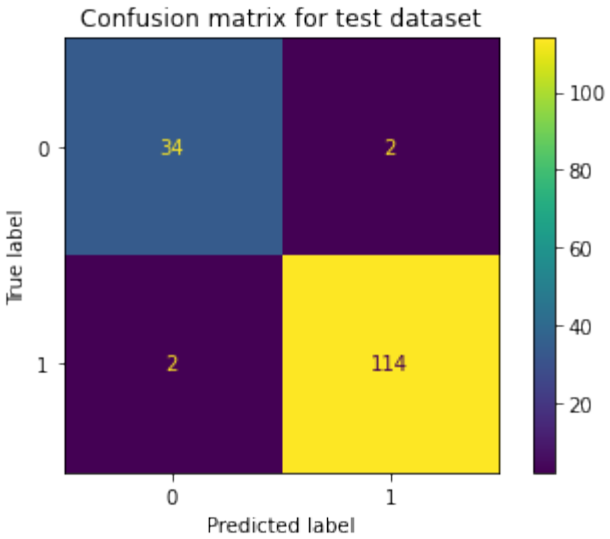
```
weighted avg     1.0000    1.0000    1.0000       604
```



In [140]:

```
print('Performance on test data:\n')
print(
    classification_report(y_test,
                          sclf.predict(X_important_test),
                          labels=[0, 1],
                          digits=4))
plot_confusion_matrix(sclf, X_important_test, y_test, labels=[0, 1])
plt.title('Confusion matrix for test dataset')
plt.show()
```

```
Performance on test data:

              precision    recall  f1-score   support

           0     0.9444    0.9444    0.9444        36
           1     0.9828    0.9828    0.9828       116

    accuracy                         0.9737       152
   macro avg     0.9636    0.9636    0.9636       152
weighted avg     0.9737    0.9737    0.9737       152
```



13 Summary

13.1 Plot the training accuracy for each estimator

In [159]:

```
estimators.append(best_lr)
est_names.append('Logistic Regression')
```

In [211]:

```
def plot_bar_acc(acc, title):
    plt.figure(figsize=(16, 8))
    ax = sns.barplot(x=est_names, y=acc)
    plt.ylim((0, 1.3))
    ax.yaxis.set_major_formatter(FuncFormatter('{0:.0%}'.format))
    totals = []
    for i in ax.patches:
        totals.append(i.get_height())

    # set individual bar lables using above list
    total = sum(totals)

    # set individual bar lables using above list
    for i, n in zip(ax.patches, range(len(train_acc))):
        # get_x pulls left or right; get_height pushes up or down
        ax.text(i.get_x() + 0.1,
                i.get_height() + .05,
                str(np.round(acc[n], 2) * 100) + '%',
                fontsize=15,
                color='black')
    plt.xlabel('estimators')
    plt.ylabel('accuracy')
    plt.title(title)
    plt.show()
```

In [212]:

```
def print_acc(train_acc, test_acc):
    for trn_acc,tst_acc, name in zip(train_acc, test_acc, est_names):
        print(name + ' training accuracy: {:.2f}'.format(trn_acc*100))
        print(name + ' test accuracy: {:.2f}\n'.format(tst_acc*100))
```

In [213]:

```
train_acc = [estimator.score(X_important_train, y_train) for estimator in estimators]
test_acc = [estimator.score(X_important_test, y_test) for estimator in estimators]
```

In [214]:

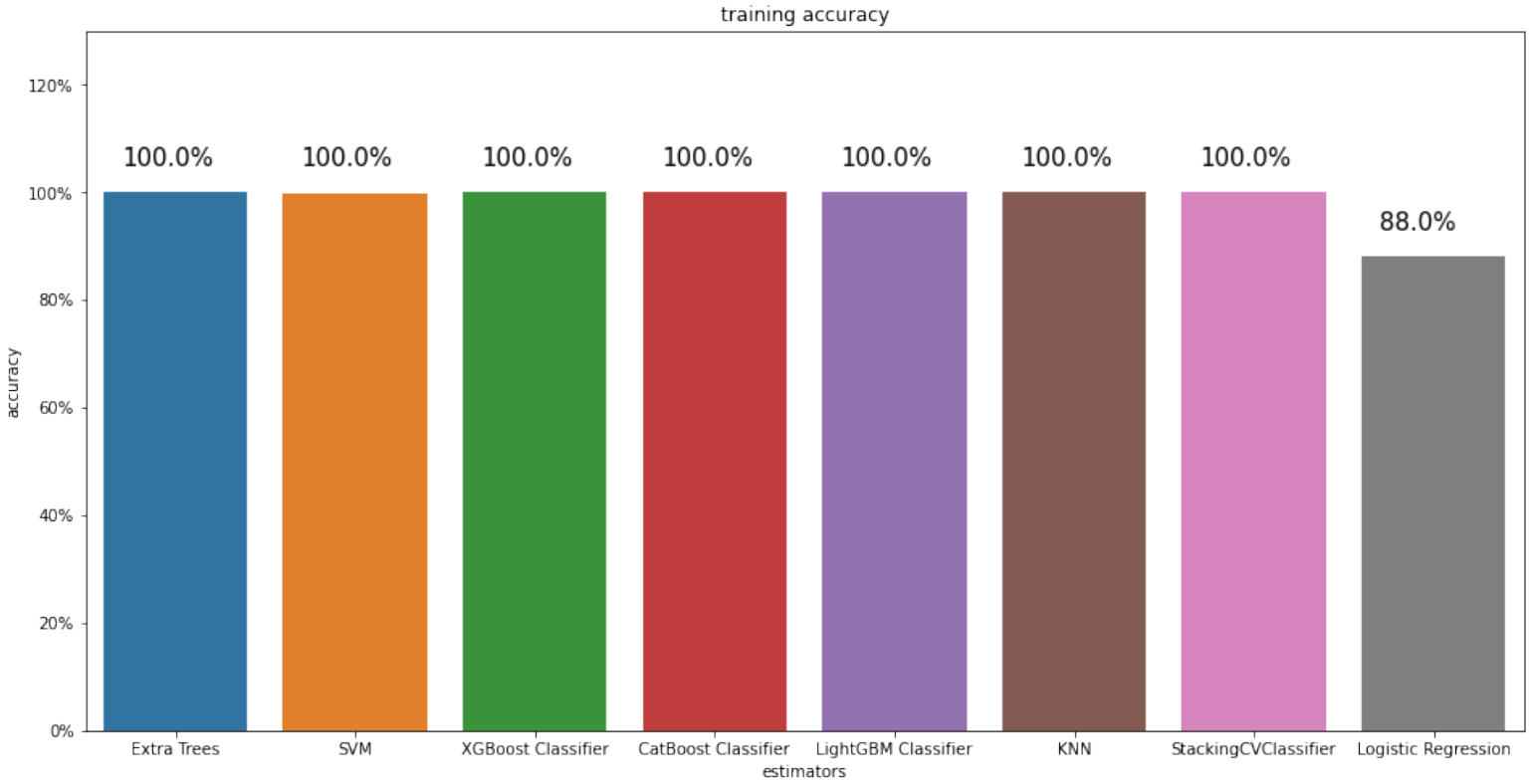
```
print_acc(train_acc, test_acc)
```

```
Extra Trees training accuracy: 100.00
Extra Trees test accuracy: 91.45

SVM training accuracy: 99.83
SVM test accuracy: 92.76

XGBoost Classifier training accuracy: 100.00
XGBoost Classifier test accuracy: 93.42

CatBoost Classifier training accuracy: 100.00
CatBoost Classifier test accuracy: 93.42

LightGBM Classifier training accuracy: 100.00
LightGBM Classifier test accuracy: 93.42
```

```
KNN training accuracy: 100.00
KNN test accuracy: 97.37

StackingCVClassifier training accuracy: 100.00
StackingCVClassifier test accuracy: 97.37

Logistic Regression training accuracy: 88.08
Logistic Regression test accuracy: 91.45
```

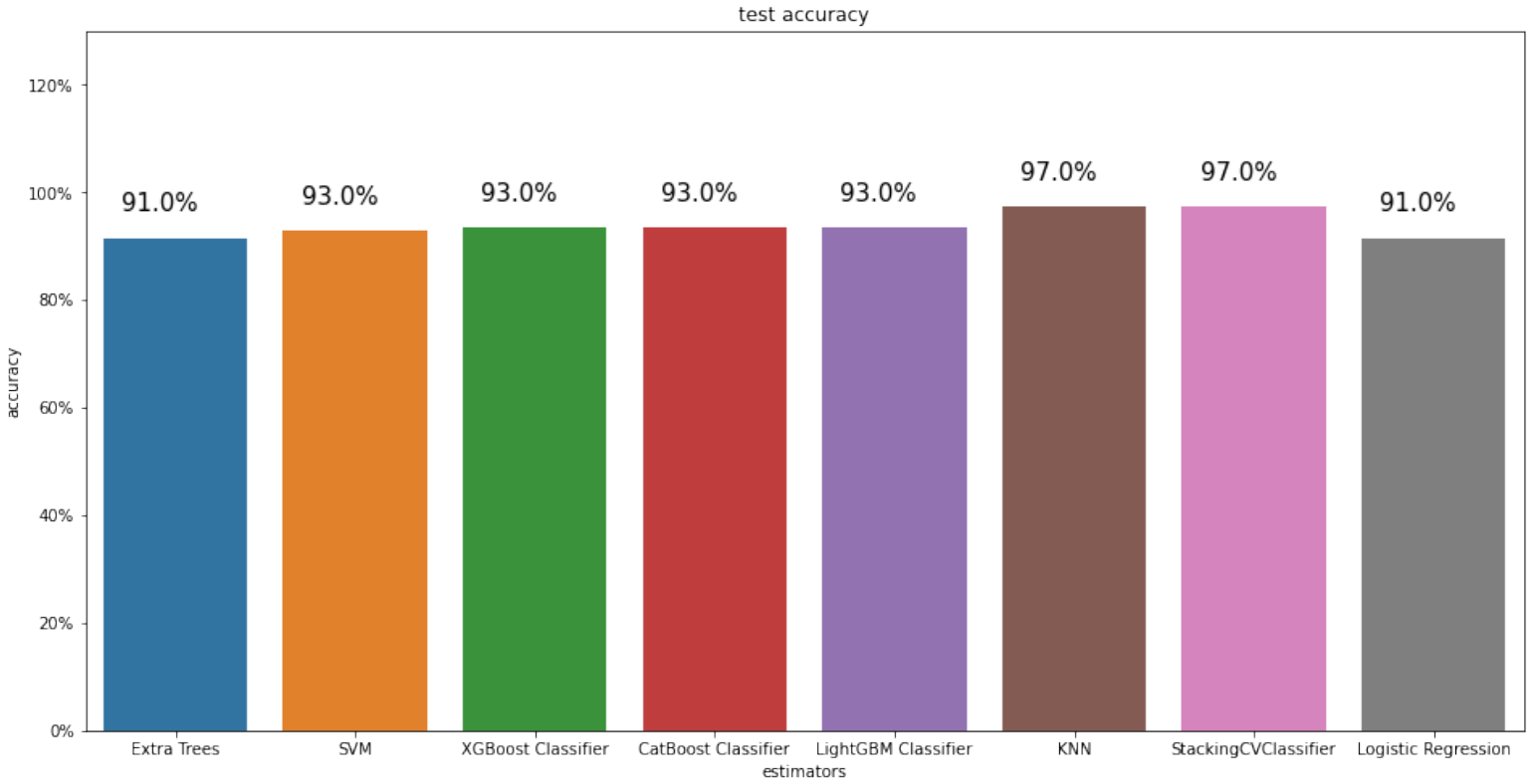
In [215]:

```
plot_bar_acc(train_acc, 'training accuracy')
```



In [216]:

```
plot_bar_acc(test_acc, 'test accuracy')
```


test accuracy
120%
100%
80%
60%
40%
20%
0%
accuracy
91.0%
93.0%
93.0%
93.0%
93.0%
97.0%
97.0%
91.0%
Extra Trees
SVM
XGBoost Classifier
CatBoost Classifier
LightGBM Classifier
KNN
StackingCVClassifier
Logistic Regression
estimators

In [ ]: