MP5

December 2, 2025

1 Machine Exercise 5

[248]:
```
import ipywidgets as widgets
from IPython.display import HTML, display

picker = widgets.ColorPicker(description="Highlight Color", value="#15a167")

def update_css(change):
    HTML(f"""
    <style>
    .highlight-global {{
        background-color: {change['new']};
    }}
    </style>
    """)

picker.observe(update_css, names='value')
display(picker)
```

```
ColorPicker(value='#15a167', description='Highlight Color')
```

This is yellow

This is black

This will follow the picker value

1.1 Part 1: Country Data from HELP International

1.1.1 1.I. Load Dataset

[226]:
```
# Source Code From : https://www.kaggle.com/datasets/rohan0301/
 ↪unsupervised-learning-on-country-data/data
# Install dependencies as needed:
# pip install kagglehub[pandas-datasets]
import kagglehub
from kagglehub import KaggleDatasetAdapter

# Set the path to the file you'd like to load
file_path = "Country-data.csv"
```

```
# Load the latest version
df = kagglehub.load_dataset(
  KaggleDatasetAdapter.PANDAS,
  "rohan0301/unsupervised-learning-on-country-data",
  file_path,
  # Provide any additional arguments like
  # sql_query or pandas_kwargs. See the
  # documentation for more information:
  # https://github.com/Kaggle/kagglehub/blob/main/README.
  ↪md#kaggledatasetadapterpandas
)

print("First 5 records:", df.head())
```

```
C:\Users\jhon\AppData\Local\Temp\ipykernel_21388\2839170658.py:11:
DeprecationWarning: Use dataset_load() instead of load_dataset(). load_dataset()
will be removed in a future version.
  df = kagglehub.load_dataset(
```

```
First 5 records:                country  child_mort  exports  health  imports
income  \
0          Afghanistan        90.2     10.0    7.58     44.9    1610
1              Albania        16.6     28.0    6.55     48.6    9930
2              Algeria        27.3     38.4    4.17     31.4   12900
3               Angola       119.0     62.3    2.85     42.9    5900
4  Antigua and Barbuda        10.3     45.5    6.03     58.9   19100

   inflation  life_expec  total_fer   gdpp
0       9.44        56.2       5.82    553
1       4.49        76.3       1.65   4090
2      16.10        76.5       2.89   4460
3      22.40        60.1       6.16   3530
4       1.44        76.8       2.13  12200
```

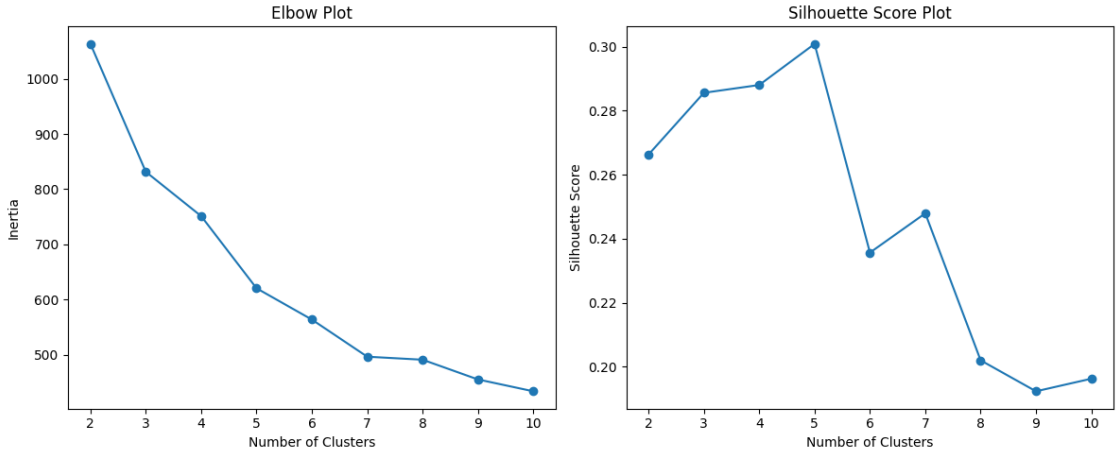
[227]:
```
# Prompt: Store Features and Target Variable
X = df.drop(columns=['country'])
Y = df['country']
```

1.1.2 1.II. K-Means

Normalize the features data using Standard Scaler. Then, perform K-means clustering on all features. Display the elbow plot (Inertia vs. no. of clusters) and the silhouette score plot. What number of clusters is recommended?

[228]:
```
# Prompt: Normalize the features of X using StandardScaler
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

```python
from sklearn.metrics import silhouette_score
import numpy as np
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

import matplotlib.pyplot as plt

# Define the range for the number of clusters
cluster_range = range(2, 11)
inertia = []
silhouette_scores = []

# Perform KMeans clustering for each number of clusters
for n_clusters in cluster_range:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)
    silhouette_scores.append(silhouette_score(X_scaled, kmeans.labels_))


def plot_elbow_and_silhouette(cluster_range, inertia, silhouette_scores):
    # Plot the elbow plot (Inertia vs. number of clusters)
    plt.figure(figsize=(12, 5))

    plt.subplot(1, 2, 1)
    plt.plot(cluster_range, inertia, marker='o')
    plt.title('Elbow Plot')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Inertia')

    # Plot the silhouette score plot
    plt.subplot(1, 2, 2)
    plt.plot(cluster_range, silhouette_scores, marker='o')
    plt.title('Silhouette Score Plot')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Silhouette Score')

    plt.tight_layout()
    plt.show()

# Call the function
plot_elbow_and_silhouette(cluster_range, inertia, silhouette_scores)
```

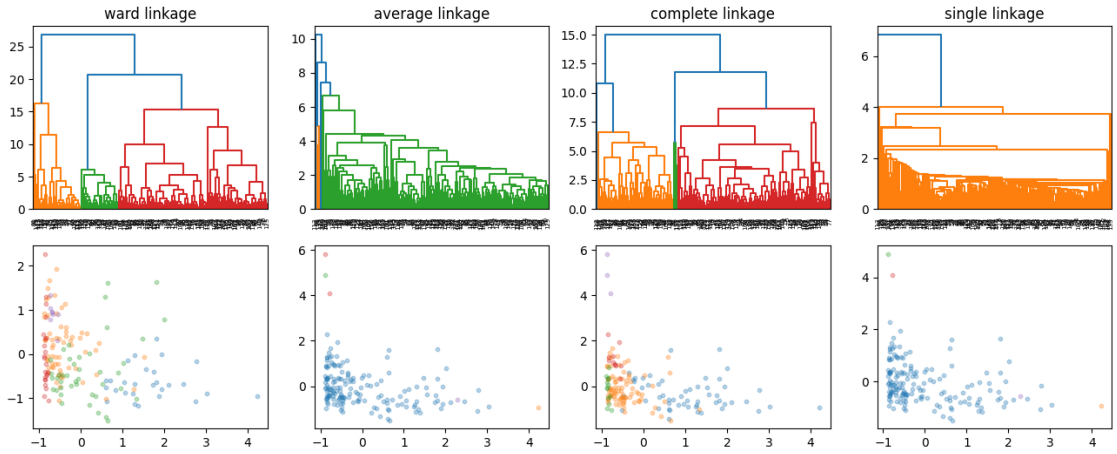


Based on the elbow plot and Silhouette Score, the recommended number of clusters is either 5 or 7. For both 5 and 7 clusters, the Silhouette Score increases, indicating that the data points are well-clustered and distinct from other clusters. The elbow plot shows the relationship between the number of clusters and the inertia. At 7 clusters, the rate of decrease in inertia slows down significantly compared to 5 clusters.

If I were to choose only one, I would select 7 clusters.

1.1.3 1.III. Hierarchical clustering

[229]:
```
# Soure Code is From AI221 Github Repository:␣
↪AI221\Clustering_Anomaly_Detect\kmeans_agglo_blobs.ipynb
from scipy.cluster.hierarchy import linkage, dendrogram, cut_tree
from time import time

fig3 = plt.figure(figsize=(16,6))
ctr = 1
t0 = time()
for method in ('ward', 'average', 'complete', 'single'):
    Z = linkage(X_scaled, method=method)
    ax = fig3.add_subplot(240 + ctr)
    ax.set_title("%s linkage" % method)
    dendrogram(Z)
    cutree = cut_tree(Z, n_clusters=7).flatten()
    ax = fig3.add_subplot(240 + ctr + 4)
    for j in range(0,5):
        ax.scatter(X_scaled[cutree == j,0], X_scaled[cutree == j,1], s=10,␣
↪alpha=0.3)
    ctr += 1

plt.show()
print(f"Elapsed Time: {time()-t0} sec")
```




```
Elapsed Time: 1.4114634990692139 sec
```

Perform hierarchical clustering on the normalized data set and compare the results of various linkage methods. Which one would you recommend? Why is this recommended clustering informative?

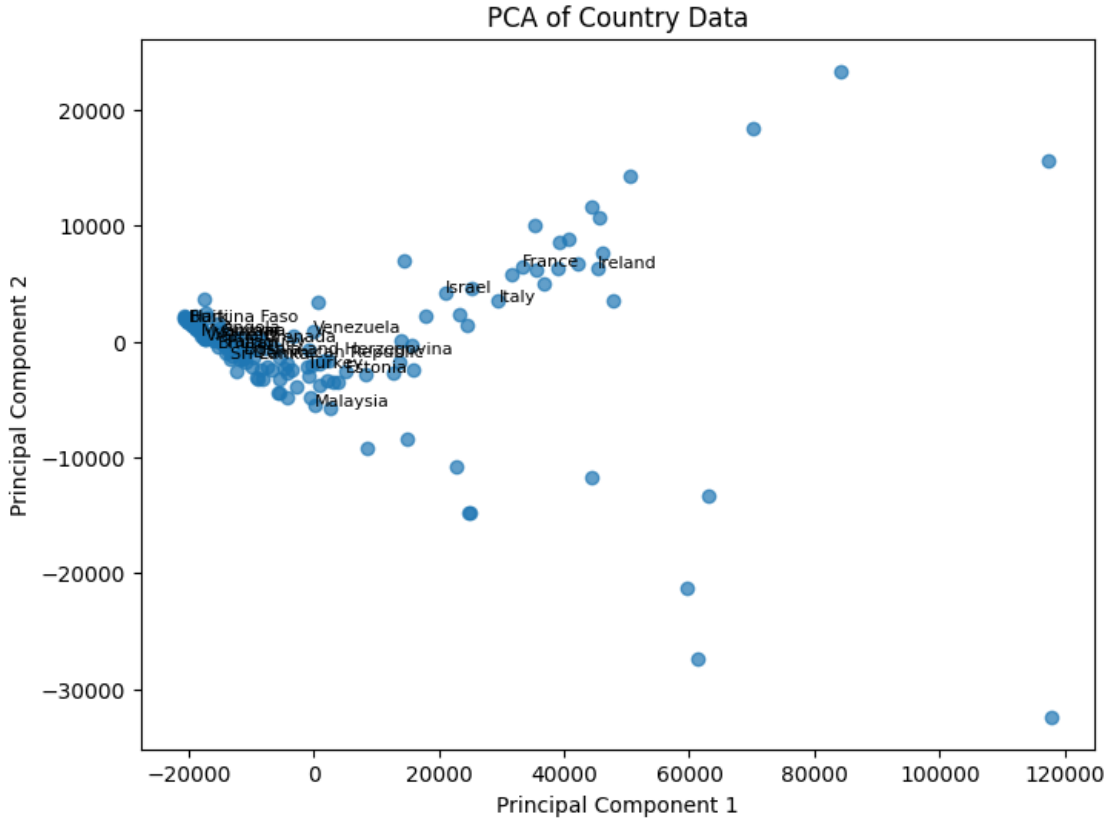
I recommend using the Ward linkage method. For this analysis, I set cut_tree to 7 clusters to align with my response to the previous item. The visualization demonstrates that the Ward linkage method is more informative compared to other methods. Ward linkage minimizes the variance within clusters, resulting in compact and well-separated clusters. This makes it particularly effective when the data has a Gaussian-like distribution or when compact clusters are desired. The method ensures that the clusters formed are homogeneous and interpretable, making it a suitable choice for this dataset.

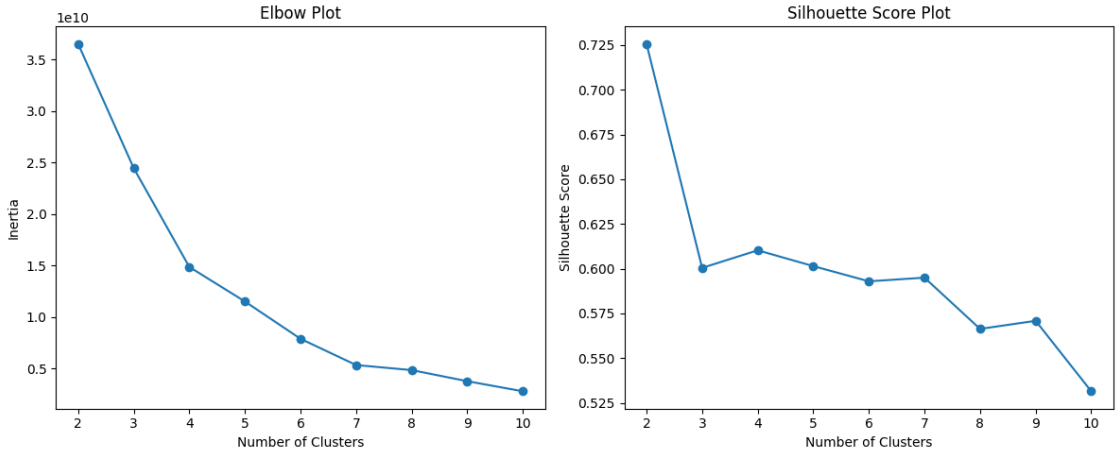
1.1.4 1.IV. PCA (2 features) + K-means

This time, perform PCA to reduce the features data into 2D. Then, perform K-means clustering on the reduced data set just as in item (a). Display the elbow plot and silhouette score plot as well.

[230]:
```
# Prompt: Perform PCA on X to reduce its dimensions to 2 components
from sklearn.decomposition import PCA
import random
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# Prompt: Display a scatter plot of the PCA-reduced data with some points␣
 ↪labeled
plt.figure(figsize=(8, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], alpha=0.7)
# Prompt: Randomly select some countries to label
selected_countries = random.sample(list(df['country']), 20)

for i, country in enumerate(df['country']):
    if country in selected_countries:
```

```
        plt.text(X_pca[i, 0], X_pca[i, 1], s=country, fontsize=8)
plt.title('PCA of Country Data')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
# Repeat KMeans clustering on PCA-reduced data
cluster_range = range(2, 11)
inertia = []
silhouette_scores = []

# Perform KMeans clustering for each number of clusters
for n_clusters in cluster_range:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(X_pca)
    inertia.append(kmeans.inertia_)
    silhouette_scores.append(silhouette_score(X_pca, kmeans.labels_))

plot_elbow_and_silhouette(cluster_range, inertia, silhouette_scores)
```





What number of clusters is recommended? You can label some countries in the 2D mapping.

I would recommend 7 clusters as it is the number where the elbow plot shows a significant decrease in inertia. Additionally, the Silhouette score for 7 clusters is higher than that of its neighboring cluster numbers, indicating well-defined and distinct clusters.

Based on the recommended no. of clusters in item (b), make your own descriptions of each cluster. What range of values of features are unique to each cluster?

To support my answer, let me get insights from the table, and plots below. I will base my descriptions on the mean values of the feature variables for each cluster. Below are the descriptions for each cluster:

- **Cluster 1**: This cluster is characterized by high health values (10.51) and low inflation (1.33). These countries likely have strong healthcare systems and stable economies.
- **Cluster 2**: Defined by the lowest income (3017.69), life expectancy (59.04), and GDP per capita (1455.76). These countries may face significant economic and health challenges.
- **Cluster 3**: Notable for the highest child mortality (130.00), inflation (104.00), and fertility rates (5.84). These countries may have developing economies with high population growth.
- **Cluster 4**: Distinguished by the highest exports (176.00), imports (156.67), life expectancy (81.43), and GDP per capita (57566.67). These countries are likely developed nations with strong trade and high standards of living.
- **Cluster 5**: Characterized by the highest income (67171.43) and lowest health values (3.28). These countries may have high wealth but face challenges in healthcare access or outcomes.
- **Cluster 0**: Represents countries with average values across most features, indicating balanced development without extremes in any specific area.
- **Cluster 6**: Defined by unique outliers or anomalies in the dataset, possibly representing countries with rare or exceptional conditions not captured by other clusters.

[231]:
```
import seaborn as sns

cluster_labels = KMeans(n_clusters=7, random_state=42).fit_predict(X_scaled)
df_with_clusters = df.copy()
```

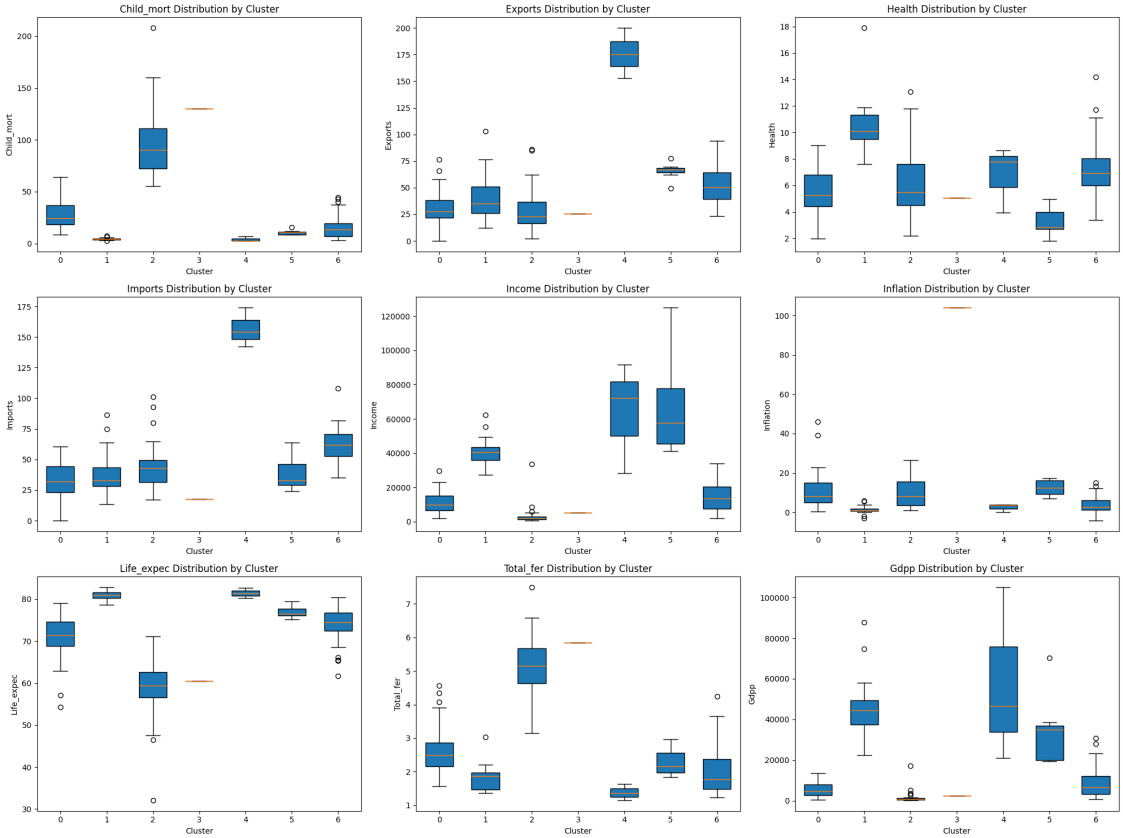
```
df_with_clusters['Cluster'] = cluster_labels  # Add cluster labels to the␣
 ↪DataFrame
cluster_summary = df_with_clusters.drop(columns=['country']).groupby('Cluster').
 ↪agg(['mean'])
print(display(cluster_summary))

# Prompt: Plot all feature distributions by cluster in the same figure
plt.figure(figsize=(20, 15))
num_features = len(df_with_clusters.columns[1:-1])  # Exclude 'country' and␣
 ↪'Cluster' columns
for i, feature in enumerate(df_with_clusters.columns[1:-1], 1):
    plt.subplot((num_features + 2) // 3, 3, i)  # Arrange subplots in a grid
    for cluster in sorted(df_with_clusters['Cluster'].unique()):
        cluster_data = df_with_clusters[df_with_clusters['Cluster'] == cluster]
        plt.boxplot(cluster_data[feature], positions=[cluster], widths=0.6,␣
 ↪patch_artist=True)
    plt.title(f'{feature.capitalize()} Distribution by Cluster')
    plt.xlabel('Cluster')
    plt.ylabel(feature.capitalize())

plt.tight_layout()
plt.show()

# Determine which cluster has the highest and lowest values for each column
highest_lowest_clusters = {}

for column in df_with_clusters.columns[1:-1]:  # Exclude 'country' and␣
 ↪'Cluster' columns
    cluster_means = df_with_clusters.groupby('Cluster')[column].mean()
    highest_cluster = cluster_means.idxmax()
    lowest_cluster = cluster_means.idxmin()
    highest_lowest_clusters[column] = {
        'highest_cluster': highest_cluster,
        'highest_value': cluster_means[highest_cluster],
        'lowest_cluster': lowest_cluster,
        'lowest_value': cluster_means[lowest_cluster]
    }

# Display the results
for feature, clusters in highest_lowest_clusters.items():
    print(f"Feature: {feature}")
    print(f"  Highest Cluster: {clusters['highest_cluster']} (Value:␣
 ↪{clusters['highest_value']:.2f})")
    print(f"  Lowest Cluster: {clusters['lowest_cluster']} (Value:␣
 ↪{clusters['lowest_value']:.2f})")
```

```
        child_mort      exports      health      imports        income  \
```

	mean	mean	mean	mean	mean
Cluster					
0	29.430435	30.621500	5.569565	33.577520	10946.956522
1	4.295652	40.730435	10.513478	38.247826	40265.217391
2	96.009524	27.988333	6.334286	43.811905	3017.690476
3	130.000000	25.300000	5.070000	17.400000	5150.000000
4	4.133333	176.000000	6.793333	156.666667	64033.333333
5	10.700000	65.557143	3.281429	38.700000	67171.428571
6	15.306667	51.824444	7.238889	62.400000	14206.888889

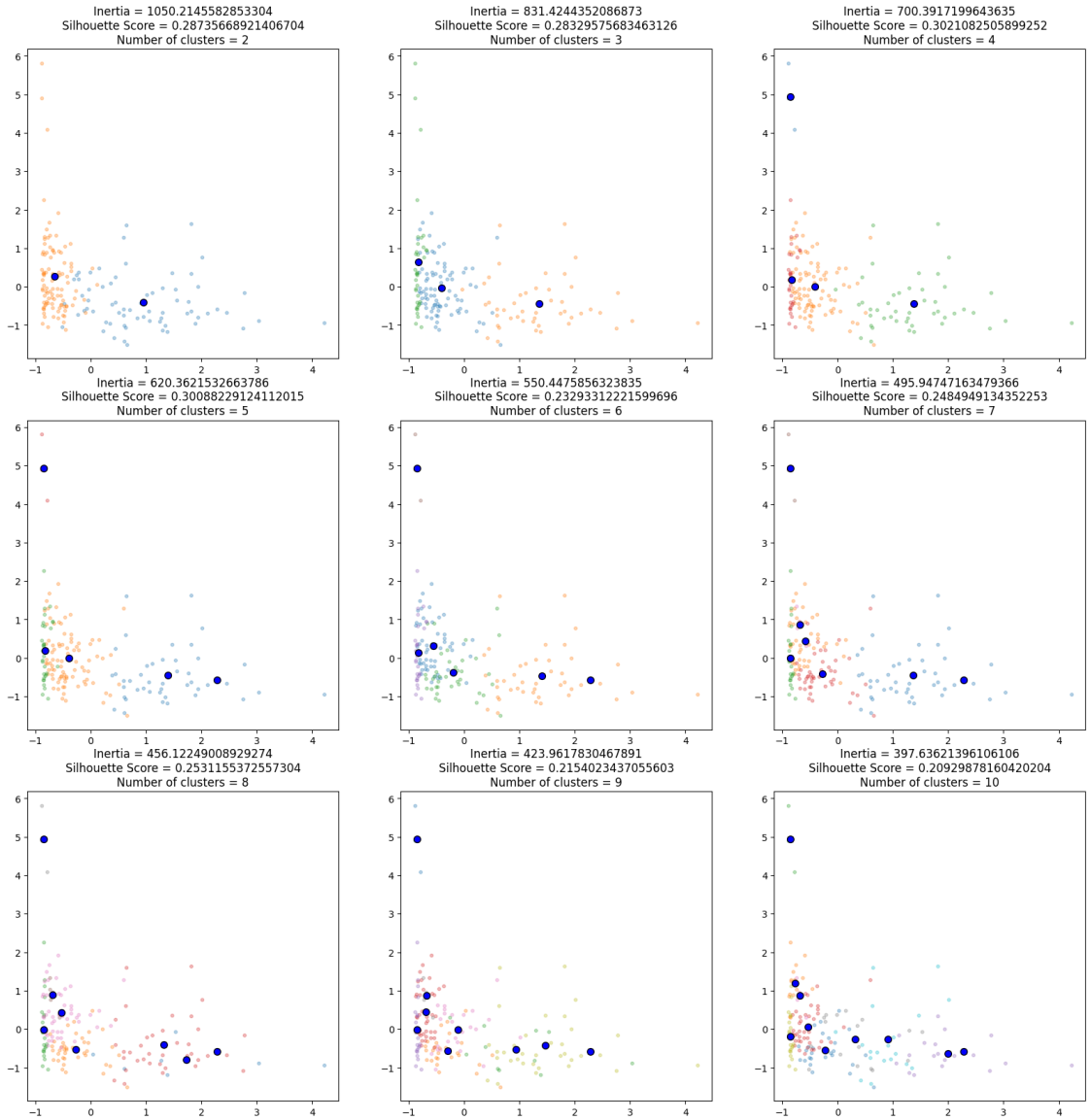
	inflation	life_expec	total_fer	gdpp
	mean	mean	mean	mean
Cluster				
0	10.601435	71.139130	2.590652	5526.000000
1	1.334913	80.891304	1.810870	45417.391304
2	9.820357	59.042857	5.129286	1455.761905
3	104.000000	60.500000	5.840000	2330.000000
4	2.468000	81.433333	1.380000	57566.666667
5	12.517143	76.928571	2.287143	34057.142857
6	3.771511	73.928889	2.001556	8703.244444

None



```
Feature: child_mort
  Highest Cluster: 3 (Value: 130.00)
  Lowest Cluster: 4 (Value: 4.13)
Feature: exports
  Highest Cluster: 4 (Value: 176.00)
  Lowest Cluster: 3 (Value: 25.30)
Feature: health
  Highest Cluster: 1 (Value: 10.51)
  Lowest Cluster: 5 (Value: 3.28)
Feature: imports
  Highest Cluster: 4 (Value: 156.67)
  Lowest Cluster: 3 (Value: 17.40)
Feature: income
  Highest Cluster: 5 (Value: 67171.43)
  Lowest Cluster: 2 (Value: 3017.69)
Feature: inflation
  Highest Cluster: 3 (Value: 104.00)
  Lowest Cluster: 1 (Value: 1.33)
Feature: life_expec
  Highest Cluster: 4 (Value: 81.43)
  Lowest Cluster: 2 (Value: 59.04)
Feature: total_fer
  Highest Cluster: 3 (Value: 5.84)
  Lowest Cluster: 4 (Value: 1.38)
Feature: gdpp
  Highest Cluster: 4 (Value: 57566.67)
  Lowest Cluster: 2 (Value: 1455.76)
```

[232]:
```
inertia_values = []
silhouette_avgs = []
fig2 = plt.figure(figsize=(20,20))
for i in range(2,11):
    kmeans = KMeans(n_clusters=i, n_init=10).fit(X_scaled)
    cluster_labels = kmeans.labels_
    centroids = kmeans.cluster_centers_
    inertia_values.append(kmeans.inertia_)
    silhouette_avg = silhouette_score(X_scaled, cluster_labels)
    silhouette_avgs.append(silhouette_avg)
    ax = fig2.add_subplot(330 + i - 1)
    for j in range(0,i):
        ax.scatter(X_scaled[cluster_labels == j,0], X_scaled[cluster_labels ==␣
↪j,1], s=10, alpha=0.3)
    ax.scatter(centroids[:,0],centroids[:,1], s=50, color='b', edgecolor='k')
    ax.set_title(f"Inertia = {kmeans.inertia_}\nSilhouette Score =␣
↪{silhouette_avg}\nNumber of clusters = {i}")
```



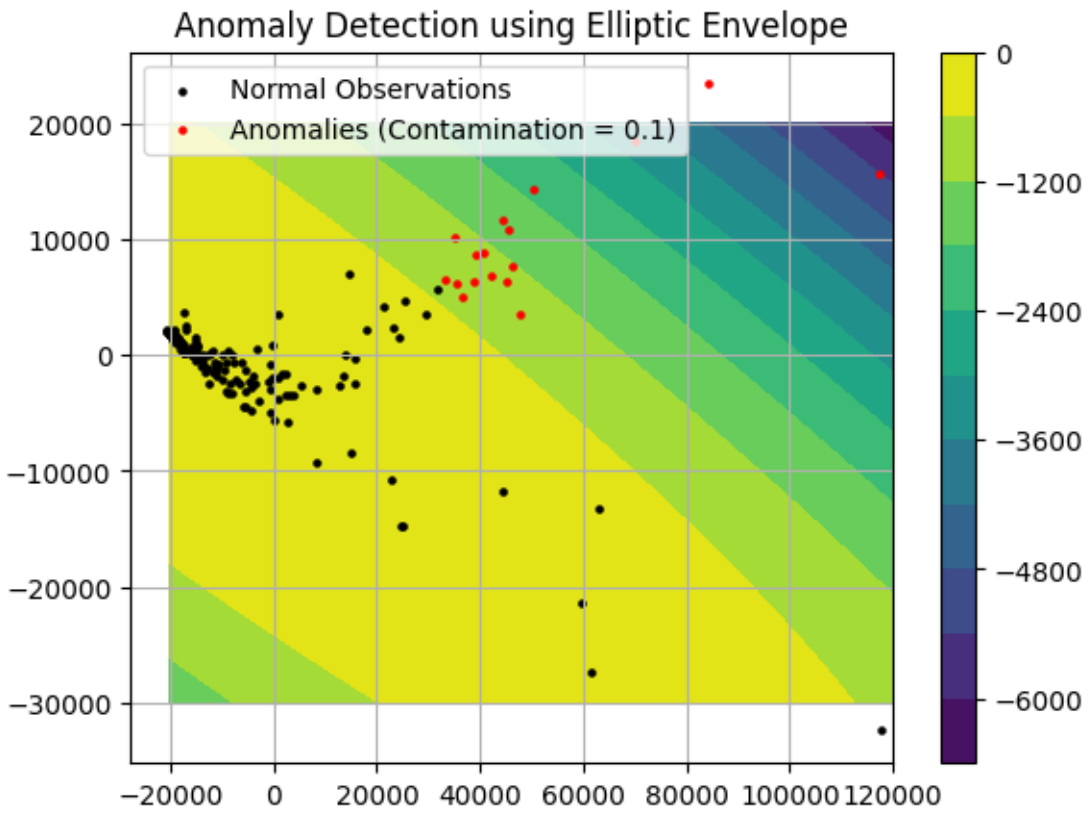
1.1.5 1.V. 2D PCA + Anomaly Detetion

Based on the 2D PCA mapping, perform anomaly detection using any method.

[233]:
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.covariance import EllipticEnvelope
from sklearn.neighbors import LocalOutlierFactor
from sklearn.neighbors import KernelDensity
from sklearn.ensemble import IsolationForest
from sklearn.datasets import make_blobs
```

```
from sklearn.svm import OneClassSVM
```

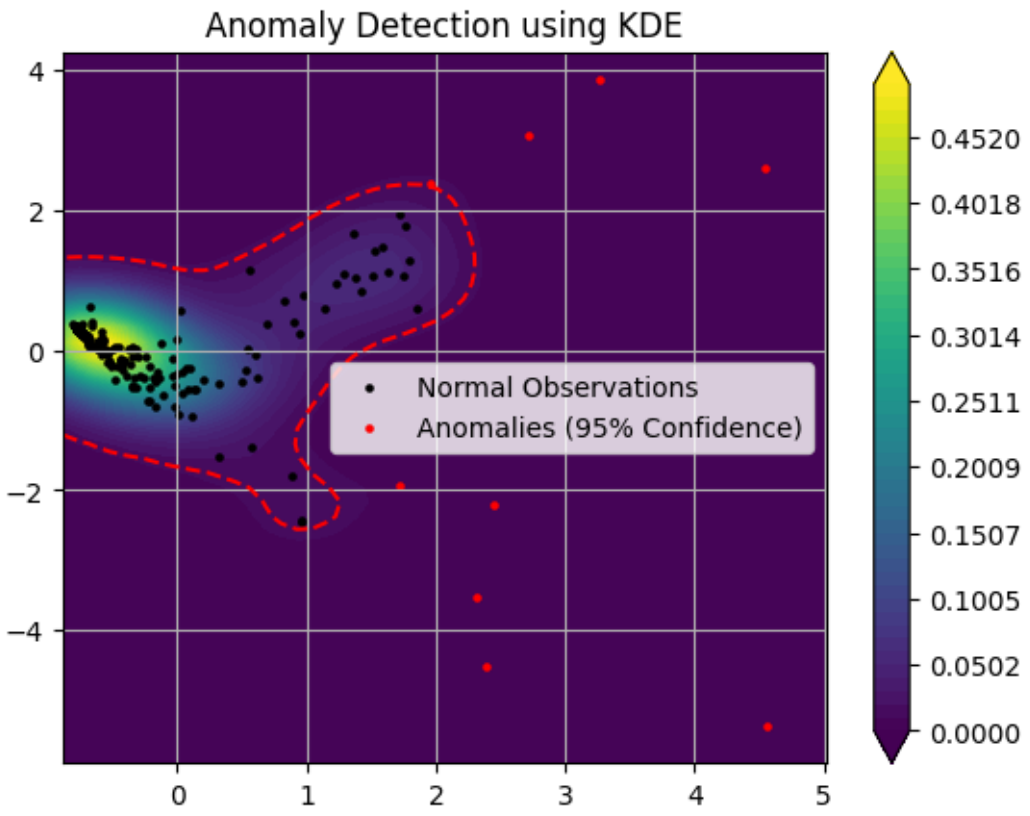
[234]:
```
# Source Code is From AI221 Github Repository:␣
↪AI221\Clustering_Anomaly_Detect\anomaly_detect_methods.ipynb
Xp, Yp = np.meshgrid(np.linspace(-20000,120000),np.linspace(-30000,20000))
XY = np.vstack([Xp.ravel(), Yp.ravel()]).T

envelope = EllipticEnvelope(random_state=0, contamination=0.1).fit(X_pca)
Zp = envelope.score_samples(XY)
Zp = Zp.reshape(Xp.shape)

# Get the anomalous data points
y_pred = envelope.predict(X_pca)
normals = X_pca[y_pred == 1,:]
anomals = X_pca[y_pred == -1,:]

cntr = plt.contourf(Xp, Yp, Zp, levels=10, cmap='viridis')
plt.scatter(normals[:,0], normals[:,1], s=5, color='k', label='Normal␣
↪Observations')
plt.scatter(anomals[:,0], anomals[:,1], s=5, color='r', label='Anomalies␣
↪(Contamination = 0.1)')
plt.title('Anomaly Detection using Elliptic Envelope')
plt.colorbar(cntr)
plt.legend()
plt.grid()
plt.show()

envelope_anomalies = df.iloc[np.where(y_pred == -1)[0]]
```



[235]:
```
# Source Code is From AI221 Github Repository:␣
 ↪AI221\Clustering_Anomaly_Detect\anomaly_detect_methods.ipynb
# Generate the KDE surface as Z
X_pca_scaled  = scaler.fit_transform(X_pca)
kde = KernelDensity(kernel='gaussian',bandwidth=0.4).fit(X_pca_scaled)


Xp, Yp = np.meshgrid(
    np.linspace(X_pca_scaled[:, 0].min() * 1.1, X_pca_scaled[:, 0].max() * 1.1,␣
 ↪50),
    np.linspace(X_pca_scaled[:, 1].min() * 1.1, X_pca_scaled[:, 1].max() * 1.1,␣
 ↪50)
)
XY = np.vstack([Xp.ravel(), Yp.ravel()]).T

Zp = np.exp(kde.score_samples(XY))
Zp = Zp.reshape(Xp.shape)

# Establish a confidence level of 95% (or 5% cutoff)
# for the UCL using the quantile of kde_scores.
```

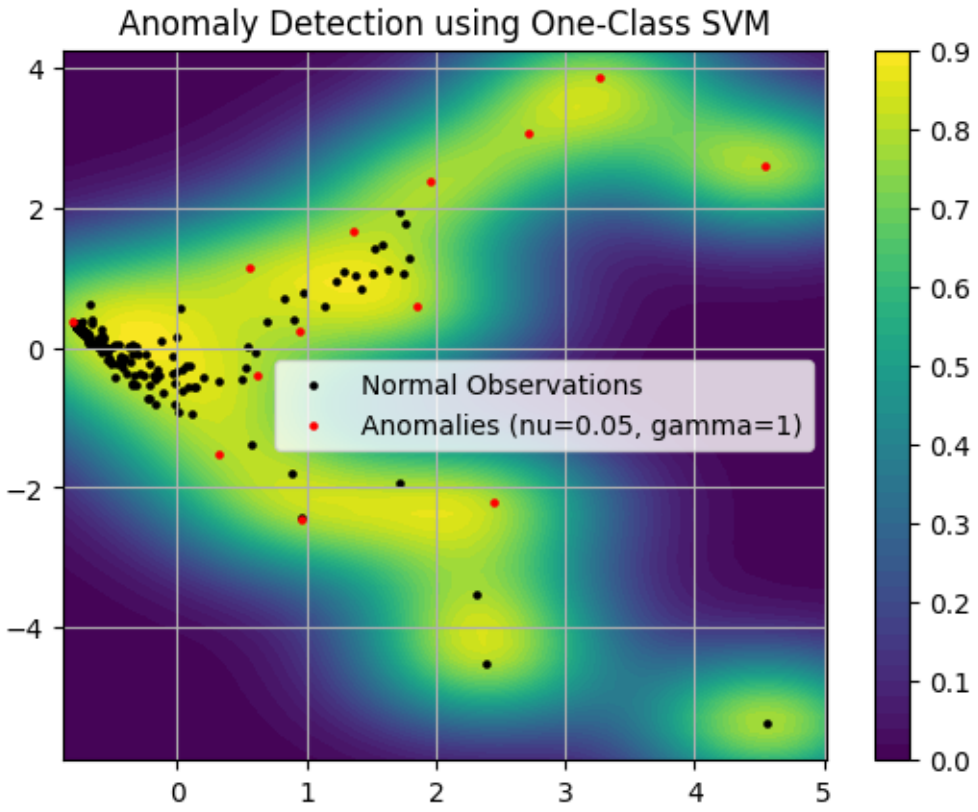
```
scores = kde.score_samples(X_pca_scaled)
threshold = np.quantile(scores,0.05)
print(f"Threshold (KDE) = {np.exp(threshold)}")

# Get the anomalous data points
normals = X_pca_scaled[scores > threshold,:]
anomals = X_pca_scaled[scores <= threshold,:]

cntr = plt.contourf(Xp, Yp, Zp, levels=np.linspace(Zp.min(), Zp.max(), 50),␣
 ↪cmap='viridis', extend='both')
plt.contour(Xp, Yp, Zp, levels=[np.exp(threshold)], colors='red',␣
 ↪linestyles='dashed', linewidths=1.5)
plt.scatter(normals[:,0], normals[:,1], s=5, color='k', label='Normal␣
 ↪Observations')
plt.scatter(anomals[:,0], anomals[:,1], s=5, color='r', label='Anomalies (95%␣
 ↪Confidence)')
plt.title('Anomaly Detection using KDE')
plt.colorbar(cntr)
plt.legend()
plt.grid()
plt.show()

kernel_density_anomalies = df.iloc[np.where(scores <= threshold)[0]]
```

```
Threshold (KDE) = 0.012569474629724105
```



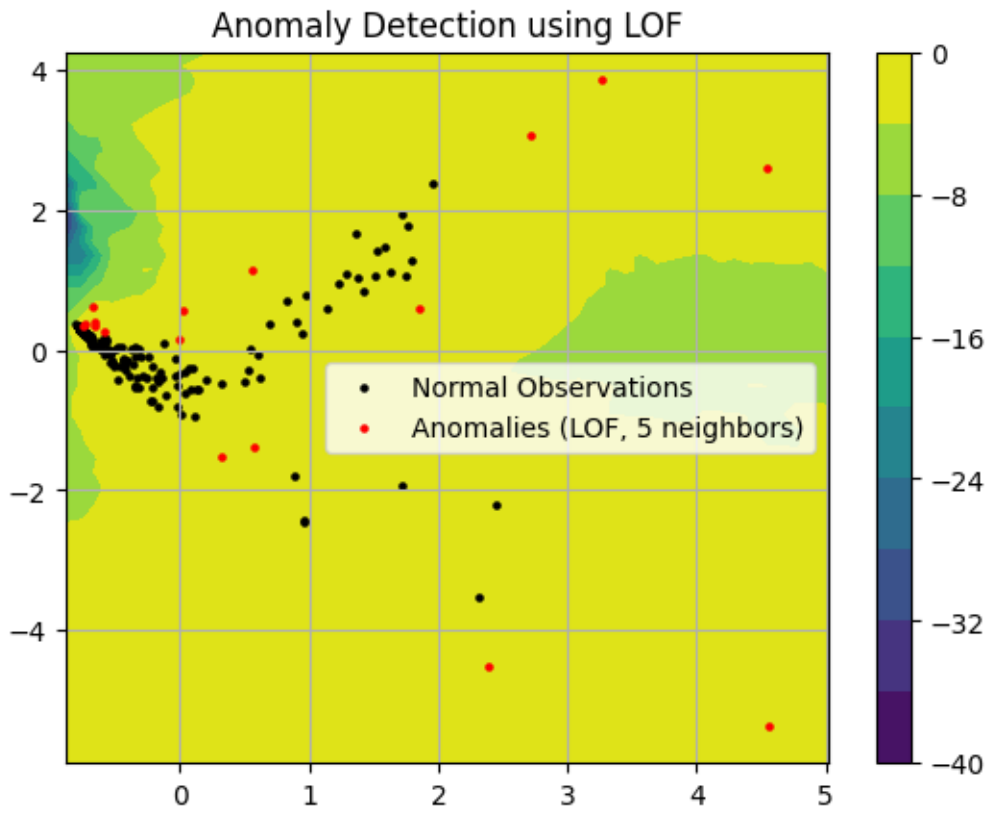
[236]:
```
# Source Code is From AI221 Github Repository:␣
 ↪AI221\Clustering_Anomaly_Detect\anomaly_detect_methods.ipynb
ocsvm = OneClassSVM(nu=0.05, gamma=1).fit(X_pca_scaled)
Zp = ocsvm.score_samples(XY)
Zp = Zp.reshape(Xp.shape)

# Get the anomalous data points
y_pred = ocsvm.predict(X_pca_scaled)
normals = X_pca_scaled[y_pred == 1,:]
anomals = X_pca_scaled[y_pred == -1,:]

cntr = plt.contourf(Xp, Yp, Zp, levels=50, cmap='viridis')
plt.scatter(normals[:,0], normals[:,1], s=5, color='k', label='Normal␣
 ↪Observations')
plt.scatter(anomals[:,0], anomals[:,1], s=5, color='r', label='Anomalies (nu=0.
 ↪05, gamma=1)')
plt.title('Anomaly Detection using One-Class SVM')
plt.colorbar(cntr)
plt.legend()
plt.grid()
```

```
plt.show()

ocsvm_anomalies = df.iloc[np.where(y_pred == -1)[0]]
```



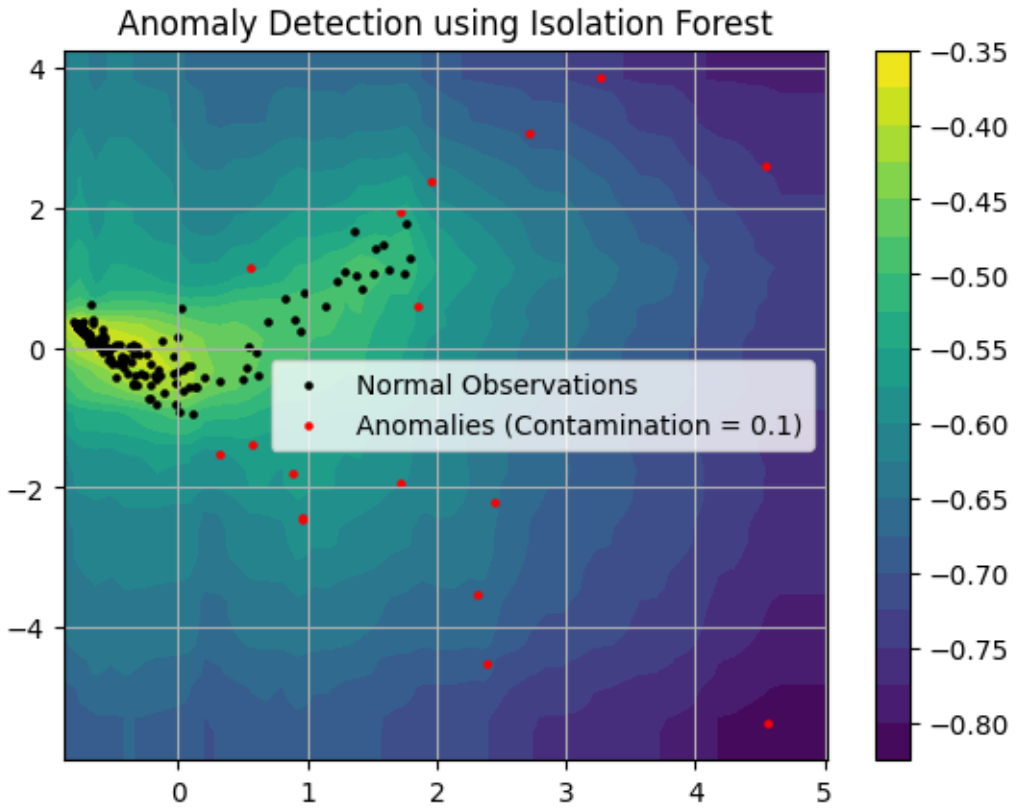
[237]:
```
# Source Code is From AI221 Github Repository:␣
 ↪AI221\Clustering_Anomaly_Detect\anomaly_detect_methods.ipynb
lof = LocalOutlierFactor(n_neighbors=5,novelty=True).fit(X_pca_scaled)
Zp = lof.score_samples(XY)
Zp = Zp.reshape(Xp.shape)

# Get the anomalous data points
y_pred = lof.predict(X_pca_scaled)
normals = X_pca_scaled[y_pred == 1,:]
anomals = X_pca_scaled[y_pred == -1,:]

cntr = plt.contourf(Xp, Yp, Zp, levels=10, cmap='viridis')
plt.scatter(normals[:,0], normals[:,1], s=5, color='k', label='Normal␣
 ↪Observations')
plt.scatter(anomals[:,0], anomals[:,1], s=5, color='r', label='Anomalies (LOF,␣
 ↪5 neighbors)')
```

```
plt.title('Anomaly Detection using LOF')
plt.colorbar(cntr)
plt.legend()
plt.grid()
plt.show()

lof_anomalies = df.iloc[np.where(y_pred == -1)[0]]
```



[238]:
```
# Source Code is From AI221 Github Repository:␣
 ↪AI221\Clustering_Anomaly_Detect\anomaly_detect_methods.ipynb
isoforest = IsolationForest(contamination=0.1).fit(X_pca_scaled)
Zp = isoforest.score_samples(XY)
Zp = Zp.reshape(Xp.shape)

# Get the anomalous data points
y_pred = isoforest.predict(X_pca_scaled)
normals = X_pca_scaled[y_pred == 1,:]
anomals = X_pca_scaled[y_pred == -1,:]

cntr = plt.contourf(Xp, Yp, Zp, levels=20, cmap='viridis')
```

```
plt.scatter(normals[:,0], normals[:,1], s=5, color='k', label='Normal␣
 ↪Observations')
plt.scatter(anomals[:,0], anomals[:,1], s=5, color='r', label='Anomalies␣
 ↪(Contamination = 0.1)')
plt.title('Anomaly Detection using Isolation Forest')
plt.colorbar(cntr)
plt.legend()
plt.grid()
plt.show()

isoforest_anomalies = df.iloc[np.where(y_pred == -1)[0]]
```



[ ]:

Which countries are deemed to be outliers and why? Which features make them outliers?

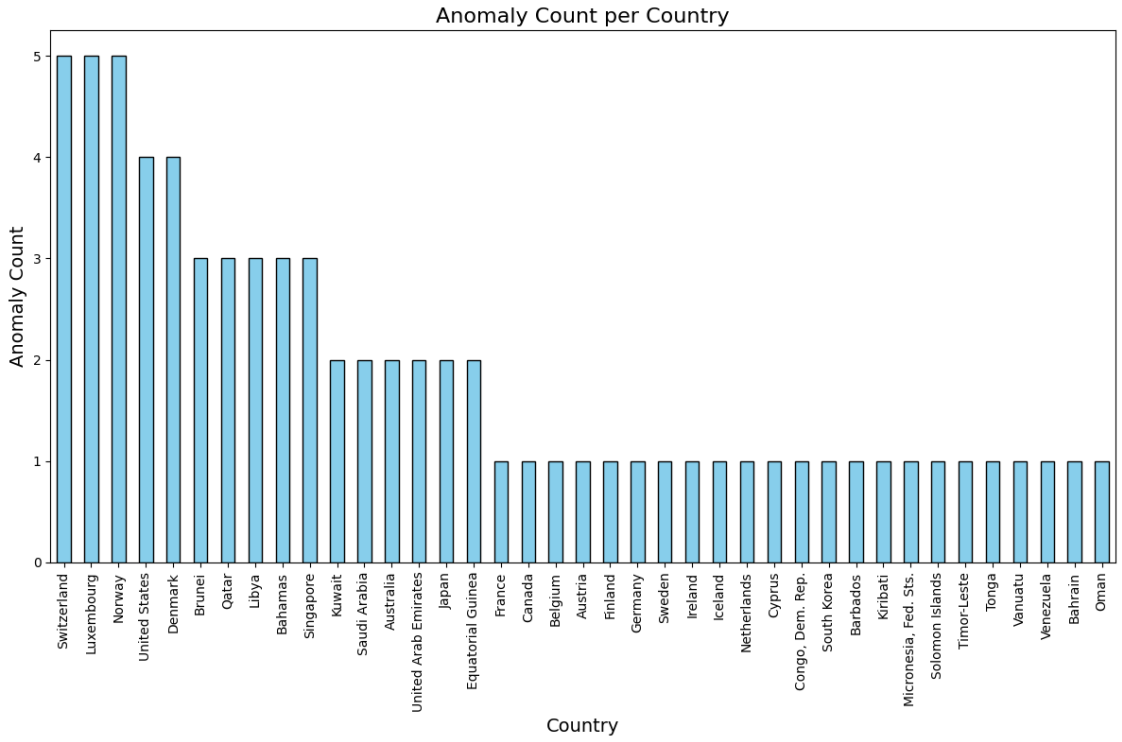
Based on the collective results from various anomaly detection models, the analysis reveals the following: Switzerland, Luxembourg, and Norway are identified as anomalies by all five models. Denmark is flagged by four models, while Japan, Brunei, Qatar, Singapore, Libya, the United States, and the Bahamas are identified by three models.

A statistical comparison between the combined anomalies from the five models and the overall

dataset highlights significant differences. The anomalies exhibit higher values in exports, imports, income, and GDP per capita (GDPP) compared to the overall dataset. This indicates that the countries identified as anomalies are developed nations.

In contrast, the majority of countries in the dataset are either underdeveloped or still developing. Alternatively, it could suggest that the anomaly countries are exceptionally successful, standing out significantly from the rest.

[239]:
```
# Prompt: Create a bar plot showing the count of anomalies detected per country␣
↪across all methods, x-axis: country, y-axis: count of anomalies
import pandas as pd

import matplotlib.pyplot as plt

# Combine all anomaly dataframes into one
all_anomalies = pd.concat([
    envelope_anomalies['country'],
    kernel_density_anomalies['country'],
    ocsvm_anomalies['country'],
    lof_anomalies['country'],
    isoforest_anomalies['country']
])

# Count the occurrences of each country
anomaly_counts = all_anomalies.value_counts()

# Plot the bar plot
plt.figure(figsize=(12, 8))
anomaly_counts.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title('Anomaly Count per Country', fontsize=16)
plt.xlabel('Country', fontsize=14)
plt.ylabel('Anomaly Count', fontsize=14)
plt.xticks(rotation=90, fontsize=10)
plt.tight_layout()
plt.show()
```



[240]:
```
all_anomalies_df = pd.concat([
    envelope_anomalies,
    kernel_density_anomalies,
    ocsvm_anomalies,
    lof_anomalies,
    isoforest_anomalies
])

# Prompt: Remove duplicate entries based on country from all_anomalies_df
all_anomalies_df = all_anomalies_df.drop_duplicates(subset=['country']).
 ↪sort_values(by='country')

print(display(all_anomalies_df))
# Prompt: Show the Mean and Standard Deviation of each feature in␣
 ↪all_anomalies_df
anomaly_stats = all_anomalies_df.describe().loc[['mean', 'std']]
print(display(anomaly_stats))
# Prompt: Show the Mean and Standard Deviation of each feature in the original␣
 ↪dataframe df
original_stats = df.describe().loc[['mean', 'std']]
print(display(original_stats))
```

```
                country  child_mort  exports  health  imports  income  \
7             Australia         4.8     19.8    8.73     20.9   41400
```

8	Austria	4.3	51.3	11.00	47.8	43200
10	Bahamas	13.8	35.0	7.89	43.7	22900
11	Bahrain	8.6	69.5	4.97	50.9	41100
13	Barbados	14.2	39.5	7.97	48.7	15300
15	Belgium	4.5	76.4	10.70	74.7	41100
23	Brunei	10.5	67.4	2.84	28.0	80600
29	Canada	5.6	29.1	11.30	31.0	40700
37	Congo, Dem. Rep.	116.0	41.1	7.91	49.6	609
42	Cyprus	3.6	50.2	5.97	57.5	33900
44	Denmark	4.1	50.5	11.40	43.6	44000
49	Equatorial Guinea	111.0	85.8	4.48	58.9	33700
53	Finland	3.0	38.7	8.95	37.4	39800
54	France	4.2	26.8	11.90	28.1	36900
58	Germany	4.2	42.3	11.60	37.1	40400
68	Iceland	2.6	53.4	9.40	43.3	38800
73	Ireland	4.2	103.0	9.19	86.5	45700
77	Japan	3.2	15.0	9.49	13.6	35800
81	Kiribati	62.7	13.3	11.30	79.9	1730
82	Kuwait	10.8	66.7	2.63	30.4	75200
89	Libya	16.6	65.6	3.88	42.1	29600
91	Luxembourg	2.8	175.0	7.77	142.0	91700
101	Micronesia, Fed. Sts.	40.0	23.5	14.20	81.0	3340
110	Netherlands	4.5	72.0	11.90	63.6	45500
114	Norway	3.2	39.7	9.48	28.5	62300
115	Oman	11.7	65.7	2.77	41.2	45300
123	Qatar	9.0	62.3	1.81	23.8	125000
128	Saudi Arabia	15.7	49.6	4.29	33.0	45400
133	Singapore	2.8	200.0	3.96	174.0	72100
136	Solomon Islands	28.1	49.3	8.55	81.2	1780
138	South Korea	4.1	49.4	6.93	46.2	30400
144	Sweden	3.0	46.2	9.63	40.7	42900
145	Switzerland	4.5	64.0	11.50	53.3	55500
149	Timor-Leste	62.6	2.2	9.12	27.8	1850
151	Tonga	17.4	12.4	5.07	60.3	4980
157	United Arab Emirates	8.6	77.7	3.66	63.6	57600
159	United States	7.3	12.4	17.90	15.8	49400
162	Vanuatu	29.2	46.6	5.25	52.7	2950
163	Venezuela	17.1	28.5	4.91	17.6	16500

	inflation	life_expec	total_fer	gdpp
7	1.160	82.0	1.93	51900
8	0.873	80.5	1.44	46900
10	-0.393	73.8	1.86	28000
11	7.440	76.0	2.16	20700
13	0.321	76.7	1.78	16000
15	1.880	80.0	1.86	44400
23	16.700	77.1	1.84	35300
29	2.870	81.3	1.63	47400

```
37      20.800        57.5       6.54     334
42       2.010        79.9       1.42   30800
44       3.220        79.5       1.87   58000
49      24.900        60.9       5.21   17100
53       0.351        80.0       1.87   46200
54       1.050        81.4       2.03   40600
58       0.758        80.1       1.39   41800
68       5.470        82.0       2.20   41900
73      -3.220        80.4       2.05   48700
77      -1.900        82.8       1.39   44500
81       1.520        60.7       3.84    1490
82      11.200        78.2       2.21   38500
89      14.200        76.1       2.41   12100
91       3.620        81.3       1.63  105000
101      3.800        65.4       3.46    2860
110      0.848        80.7       1.79   50300
114      5.950        81.0       1.95   87800
115     15.600        76.1       2.90   19300
123      6.980        79.5       2.07   70300
128     17.200        75.1       2.96   19300
133     -0.046        82.7       1.15   46600
136      6.810        61.7       4.24    1290
138      3.160        80.1       1.23   22100
144      0.991        81.5       1.98   52100
145      0.317        82.2       1.52   74600
149     26.500        71.1       6.23    3600
151      3.680        69.9       3.91    3550
157     12.500        76.5       1.87   35000
159      1.220        78.7       1.93   48400
162      2.620        63.0       3.50    2970
163     45.900        75.4       2.47   13500

None

      child_mort    exports    health    imports        income  inflation  \
mean   17.541026  54.279487  8.005128  51.282051  39408.692308   6.893846
std    26.812622  38.638883  3.608391  31.583556  26523.865636   9.778181

      life_expec  total_fer          gdpp
mean   76.123077   2.454359  35158.820513
std     7.033995   1.286196  24695.282429

None

      child_mort    exports    health    imports        income  inflation  \
mean   38.270060  41.108976  6.815689  46.890215  17144.688623   7.781832
std    40.328931  27.412010  2.746837  24.209589  19278.067698  10.570704

      life_expec  total_fer          gdpp
mean   70.555689   2.947964  12964.155689
```

```
std      8.893172    1.513848   18328.704809
```

None

1.2 2. Early-Stage Diabetes Risk Prediction

1.2.1 2.I. Load Dataset

[241]:
```
# code source: https://archive.ics.uci.edu/dataset/529/
  ↪early+stage+diabetes+risk+prediction+dataset
from ucimlrepo import fetch_ucirepo

# fetch dataset
early_stage_diabetes_risk_prediction = fetch_ucirepo(id=529)

# data (as pandas dataframes)
X = early_stage_diabetes_risk_prediction.data.features
y = early_stage_diabetes_risk_prediction.data.targets

# metadata
print(early_stage_diabetes_risk_prediction.metadata)

# variable information
print(early_stage_diabetes_risk_prediction.variables)
```

```
{'uci_id': 529, 'name': 'Early Stage Diabetes Risk Prediction',
'repository_url': 'https://archive.ics.uci.edu/dataset/529/early+stage+diabetes+
risk+prediction+dataset', 'data_url':
'https://archive.ics.uci.edu/static/public/529/data.csv', 'abstract': 'This
dataset contains the sign and symptpom data of newly diabetic or would be
diabetic patient. ', 'area': 'Computer Science', 'tasks': ['Classification'],
'characteristics': ['Multivariate'], 'num_instances': 520, 'num_features': 16,
'feature_types': ['Categorical', 'Integer'], 'demographics': ['Age', 'Gender'],
'target_col': ['class'], 'index_col': None, 'has_missing_values': 'no',
'missing_values_symbol': None, 'year_of_dataset_creation': 2020, 'last_updated':
'Mon Mar 04 2024', 'dataset_doi': '10.24432/C5VG8H', 'creators': [],
'intro_paper': {'ID': 397, 'type': 'NATIVE', 'title': 'Likelihood Prediction of
Diabetes at Early Stage Using Data Mining Techniques', 'authors': 'M. M. F.
Islam, Rahatara Ferdousi, Sadikur Rahman, Humayra Yasmin Bushra', 'venue':
'Computer Vision and Machine Intelligence in Medical Image Analysis', 'year':
2019, 'journal': None, 'DOI': '10.1007/978-981-13-8798-2_12', 'URL': 'https://ww
w.semanticscholar.org/paper/9329dec57c5f13f195220ffa7077fd0029983f07', 'sha':
None, 'corpus': None, 'arxiv': None, 'mag': None, 'acl': None, 'pmid': None,
'pmcid': None}, 'additional_info': {'summary': 'This has been col-\r\nlected
using direct questionnaires from the patients of Sylhet Diabetes\r\nHospital in
Sylhet, Bangladesh and approved by a doctor.', 'purpose': None, 'funded_by':
None, 'instances_represent': None, 'recommended_data_splits': None,
'sensitive_data': None, 'preprocessing_description': None, 'variable_info': 'Age
1.20-65\t\t\r\nSex 1. Male, 2.Female\t\t\r\nPolyuria 1.Yes,
```

2.No.\t\t\r\nPolydipsia 1.Yes, 2.No.\t\t\r\nsudden weight loss 1.Yes,
2.No.\t\t\r\nweakness 1.Yes, 2.No.\t\t\r\nPolyphagia 1.Yes, 2.No.\t\t\r\nGenital
thrush 1.Yes, 2.No.\t\t\r\nvisual blurring 1.Yes, 2.No.\t\t\r\nItching 1.Yes,
2.No.\t\t\r\nIrritability 1.Yes, 2.No.\t\t\r\ndelayed healing 1.Yes,
2.No.\t\t\r\npartial paresis 1.Yes, 2.No.\t\t\r\nmuscle sti\xObness 1.Yes,
2.No.\t\t\r\nAlopecia 1.Yes, 2.No.\t\t\r\nObesity 1.Yes, 2.No.\t\t\r\nClass
1.Positive, 2.Negative.\t\t\r\n', 'citation': None}}

	name	role	type	demographic	description	units	\
0	age	Feature	Integer	Age	None	None	
1	gender	Feature	Categorical	Gender	None	None	
2	polyuria	Feature	Binary	None	None	None	
3	polydipsia	Feature	Binary	None	None	None	
4	sudden_weight_loss	Feature	Binary	None	None	None	
5	weakness	Feature	Binary	None	None	None	
6	polyphagia	Feature	Binary	None	None	None	
7	genital_thrush	Feature	Binary	None	None	None	
8	visual_blurring	Feature	Binary	None	None	None	
9	itching	Feature	Binary	None	None	None	
10	irritability	Feature	Binary	None	None	None	
11	delayed_healing	Feature	Binary	None	None	None	
12	partial_paresis	Feature	Binary	None	None	None	
13	muscle_stiffness	Feature	Binary	None	None	None	
14	alopecia	Feature	Binary	None	None	None	
15	obesity	Feature	Binary	None	None	None	
16	class	Target	Binary	None	None	None	

	missing_values
0	no
1	no
2	no
3	no
4	no
5	no
6	no
7	no
8	no
9	no
10	no
11	no
12	no
13	no
14	no
15	no
16	no

1.2.2 2.II. Encoding, and Dataset Split

Item 2.a. Make the necessary encoding for categorical inputs. Split the data into 80% Training and 20% Testing with stratification.

[242]:
```
# Encode categorical variables
from sklearn.calibration import LabelEncoder


X = pd.get_dummies(X, drop_first=True)
# Encode y

le = LabelEncoder()
y = le.fit_transform(y)

# Split the dataset into training and testing sets (stratify = y)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42, stratify=y)
```

```
c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\preprocessing\_label.py:110: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

1.2.3 2.III. Best Model Using

Item 2.b. Using Optuna, find the best model between the MLP Classifier, Random Forest Classifier, XGBoost Classifier, Logistic Regression, Naïve Bayes Classifier, SVM Classifier (SVC), and kNN Classifier. Set Optuna to maximize the 10-fold cross-validation score (cross_val_score). You are free to design the search space for hyper-parameters in these models.

[243]:
```
import optuna
from sklearn.model_selection import cross_val_score
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, f1_score

def objective(trial):
    classifier_name = trial.suggest_categorical("classifier",
                                                ["MLP", "RandomForest",␣
 ↪"XGBoost", "LogisticRegression", "NaiveBayes", "SVM", "kNN"])
```

```
    if classifier_name == "MLP":
        hidden_layer_sizes_choices = ["(50,)", "(100,)", "(50, 50)"]
        params = {
            "hidden_layer_sizes": eval(trial.
↪suggest_categorical("hidden_layer_sizes", hidden_layer_sizes_choices)),
            "activation": trial.suggest_categorical("activation", ["relu",␣
↪"tanh"]),
            "solver": trial.suggest_categorical("solver", ["adam", "sgd"]),
            "alpha": trial.suggest_float("alpha", 1e-5, 1e-1, log=True),
        }
        model = MLPClassifier(**params, max_iter=1000, random_state=42,␣
↪early_stopping=True)

    elif classifier_name == "RandomForest":
        params = {
            "n_estimators": trial.suggest_int("n_estimators", 50, 300, step=50),
            "max_depth": trial.suggest_int("max_depth", 3, 20),
            "min_samples_split": trial.suggest_int("min_samples_split", 2, 10),
        }
        model = RandomForestClassifier(**params, random_state=42)

    elif classifier_name == "XGBoost":
        params = {
            "max_depth": trial.suggest_int("max_depth", 3, 15),
            "learning_rate": trial.suggest_float("learning_rate", 0.01, 0.3,␣
↪log=True),
            "n_estimators": trial.suggest_int("n_estimators", 50, 300, step=50),
        }
        model = XGBClassifier(**params, eval_metric="logloss", random_state=42)

    elif classifier_name == "LogisticRegression":
        params = {
            "C": trial.suggest_float("C", 1e-3, 1e3, log=True),
            "solver": trial.suggest_categorical("k-solver", ["liblinear",␣
↪"lbfgs"]),
        }
        model = LogisticRegression(C=params["C"], solver=params["solver"],␣
↪max_iter=1000, random_state=42)

    elif classifier_name == "NaiveBayes":
        model = GaussianNB()

    elif classifier_name == "SVM":
        params = {
            "C": trial.suggest_float("C", 1e-3, 1e3, log=True),
```

```
            "kernel": trial.suggest_categorical("kernel", ["linear", "rbf",␣
 ↪"poly"]),
        }
        model = SVC(**params, random_state=42)

    elif classifier_name == "kNN":
        params = {
            "n_neighbors": trial.suggest_int("n_neighbors", 3, 15),
            "weights": trial.suggest_categorical("weights", ["uniform",␣
 ↪"distance"]),
        }
        model = KNeighborsClassifier(**params)

    # Perform 10-fold cross-validation
    score = cross_val_score(model, X_train, y_train, cv=10, scoring="accuracy").
 ↪mean()
    return score

# Create and run the Optuna study
study = optuna.create_study(direction="maximize", sampler=optuna.samplers.
 ↪TPESampler(seed=42))
study.optimize(objective, n_trials=500, show_progress_bar=True)

# Get the best model and hyperparameters
best_params = study.best_params
best_classifier = best_params.pop("classifier")

if best_classifier == "MLP":
    hidden_layer_sizes = eval(best_params["hidden_layer_sizes"])
    best_model = MLPClassifier(**best_params,  max_iter=1000, random_state=42,␣
 ↪early_stopping=True)
elif best_classifier == "RandomForest":
    best_model = RandomForestClassifier(**best_params, random_state=42)
elif best_classifier == "XGBoost":
    best_model = XGBClassifier(**best_params, eval_metric="logloss",␣
 ↪random_state=42)
elif best_classifier == "LogisticRegression":
    best_model = LogisticRegression(C=best_params["C"],␣
 ↪solver=best_params["k-solver"], random_state=42)
elif best_classifier == "NaiveBayes":
    best_model = GaussianNB()
elif best_classifier == "SVM":
    best_model = SVC(**best_params, random_state=42)
elif best_classifier == "kNN":
    best_model = KNeighborsClassifier(**best_params)
```

```python
# Train the best model on the training data
best_model.fit(X_train, y_train)

# Evaluate on the test data
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average="weighted")

print(f"Best Classifier: {best_classifier}")
print(f"Best Hyperparameters: {best_params}")
print(f"Test Accuracy: {accuracy}")
print(f"Test F1-Score: {f1}")
```

```
[I 2025-12-02 14:14:35,183] A new study created in memory with name: no-
name-7b2f9dd6-3004-4566-adfd-326b6a5ea5b9

  0%|          | 0/500 [00:00<?, ?it/s]

[I 2025-12-02 14:14:37,268] Trial 0 finished with value: 0.9663182346109174 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 300, 'max_depth': 13,
'min_samples_split': 8}. Best is trial 0 with value: 0.9663182346109174.
[I 2025-12-02 14:14:38,722] Trial 1 finished with value: 0.9736353077816492 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 200, 'max_depth': 10,
'min_samples_split': 4}. Best is trial 1 with value: 0.9736353077816492.
[I 2025-12-02 14:14:38,774] Trial 2 finished with value: 0.6153310104529617 and
parameters: {'classifier': 'SVM', 'C': 1.2173252504194043, 'kernel': 'poly'}.
Best is trial 1 with value: 0.9736353077816492.
[I 2025-12-02 14:14:38,807] Trial 3 finished with value: 0.9207317073170733 and
parameters: {'classifier': 'LogisticRegression', 'C': 12.746711578215052,
'k-solver': 'liblinear'}. Best is trial 1 with value: 0.9736353077816492.
[I 2025-12-02 14:14:40,035] Trial 4 finished with value: 0.9664924506387921 and
parameters: {'classifier': 'XGBoost', 'max_depth': 10, 'learning_rate':
0.01875220945578641, 'n_estimators': 300}. Best is trial 1 with value:
0.9736353077816492.
[I 2025-12-02 14:14:40,439] Trial 5 finished with value: 0.9711962833914054 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 50, 'max_depth': 8,
'min_samples_split': 5}. Best is trial 1 with value: 0.9736353077816492.
[I 2025-12-02 14:14:40,822] Trial 6 finished with value: 0.9663182346109174 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 50, 'max_depth': 20,
'min_samples_split': 8}. Best is trial 1 with value: 0.9736353077816492.
[I 2025-12-02 14:14:41,927] Trial 7 finished with value: 0.9592915214866435 and
parameters: {'classifier': 'XGBoost', 'max_depth': 7, 'learning_rate':
0.014830392684568025, 'n_estimators': 300}. Best is trial 1 with value:
0.9736353077816492.
[I 2025-12-02 14:14:42,059] Trial 8 finished with value: 0.9254936120789778 and
parameters: {'classifier': 'SVM', 'C': 210.5118051960872, 'kernel': 'poly'}.
Best is trial 1 with value: 0.9736353077816492.
[I 2025-12-02 14:14:42,645] Trial 9 finished with value: 0.9423344947735192 and
parameters: {'classifier': 'XGBoost', 'max_depth': 4, 'learning_rate':
```

```
0.011128194768838964, 'n_estimators': 200}. Best is trial 1 with value:
0.9736353077816492.
[I 2025-12-02 14:14:42,812] Trial 10 finished with value: 0.7329268292682926 and
parameters: {'classifier': 'MLP', 'hidden_layer_sizes': '(50,)', 'activation':
'tanh', 'solver': 'adam', 'alpha': 0.00021206551805691242}. Best is trial 1 with
value: 0.9736353077816492.
[I 2025-12-02 14:14:42,853] Trial 11 finished with value: 0.8922183507549362 and
parameters: {'classifier': 'kNN', 'n_neighbors': 3, 'weights': 'uniform'}. Best
is trial 1 with value: 0.9736353077816492.
[I 2025-12-02 14:14:42,879] Trial 12 finished with value: 0.8747967479674796 and
parameters: {'classifier': 'NaiveBayes'}. Best is trial 1 with value:
0.9736353077816492.
[I 2025-12-02 14:14:43,255] Trial 13 finished with value: 0.9783972125435539 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 50, 'max_depth': 13,
'min_samples_split': 3}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:44,383] Trial 14 finished with value: 0.9760162601626016 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 150, 'max_depth': 14,
'min_samples_split': 2}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:45,115] Trial 15 finished with value: 0.9760162601626016 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 100, 'max_depth': 15,
'min_samples_split': 2}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:45,296] Trial 16 finished with value: 0.6176538908246225 and
parameters: {'classifier': 'MLP', 'hidden_layer_sizes': '(50, 50)',
'activation': 'relu', 'solver': 'sgd', 'alpha': 0.0852297911668136}. Best is
trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:45,365] Trial 17 finished with value: 0.6152729384436701 and
parameters: {'classifier': 'LogisticRegression', 'C': 0.0015791477781380012,
'k-solver': 'lbfgs'}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:45,396] Trial 18 finished with value: 0.9042392566782812 and
parameters: {'classifier': 'kNN', 'n_neighbors': 15, 'weights': 'distance'}.
Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:45,424] Trial 19 finished with value: 0.8747967479674796 and
parameters: {'classifier': 'NaiveBayes'}. Best is trial 13 with value:
0.9783972125435539.
[I 2025-12-02 14:14:46,496] Trial 20 finished with value: 0.9760162601626016 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 150, 'max_depth': 17,
'min_samples_split': 2}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:47,286] Trial 21 finished with value: 0.9760162601626016 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 100, 'max_depth': 14,
'min_samples_split': 2}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:48,046] Trial 22 finished with value: 0.9760162601626016 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 100, 'max_depth': 15,
'min_samples_split': 2}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:48,793] Trial 23 finished with value: 0.9712543554006968 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 100, 'max_depth': 17,
'min_samples_split': 4}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:49,899] Trial 24 finished with value: 0.9783972125435539 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 150, 'max_depth': 12,
```

```
'min_samples_split': 3}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:51,003] Trial 25 finished with value: 0.9736353077816492 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 150, 'max_depth': 12,
'min_samples_split': 4}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:52,479] Trial 26 finished with value: 0.954239256678281 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 200, 'max_depth': 11,
'min_samples_split': 10}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:52,511] Trial 27 finished with value: 0.6153310104529617 and
parameters: {'classifier': 'LogisticRegression', 'C': 0.0011997072580708838,
'k-solver': 'liblinear'}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:52,544] Trial 28 finished with value: 0.9185830429732869 and
parameters: {'classifier': 'kNN', 'n_neighbors': 9, 'weights': 'distance'}. Best
is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:54,323] Trial 29 finished with value: 0.9783972125435539 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 250, 'max_depth': 17,
'min_samples_split': 3}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:54,647] Trial 30 finished with value: 0.7545876887340303 and
parameters: {'classifier': 'MLP', 'hidden_layer_sizes': '(100,)', 'activation':
'tanh', 'solver': 'adam', 'alpha': 1.0065657415492051e-05}. Best is trial 13
with value: 0.9783972125435539.
[I 2025-12-02 14:14:56,456] Trial 31 finished with value: 0.9783972125435539 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 250, 'max_depth': 17,
'min_samples_split': 3}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:58,235] Trial 32 finished with value: 0.9736353077816492 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 250, 'max_depth': 19,
'min_samples_split': 4}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:14:58,296] Trial 33 finished with value: 0.6153310104529617 and
parameters: {'classifier': 'SVM', 'C': 0.06563475772795334, 'kernel': 'rbf'}.
Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:15:00,113] Trial 34 finished with value: 0.9783972125435539 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 250, 'max_depth': 17,
'min_samples_split': 3}. Best is trial 13 with value: 0.9783972125435539.
[I 2025-12-02 14:15:01,933] Trial 35 finished with value: 0.9783972125435539 and
parameters: {'classifier': 'RandomForest', 'n_estimators': 250, 'max_depth': 18,
'min_samples_split': 3}. Best is trial 13 with value: 0.9783972125435539.
[W 2025-12-02 14:15:01,954] Trial 36 failed with parameters: {'classifier':
'NaiveBayes'} because of the following error: KeyboardInterrupt().
Traceback (most recent call last):
  File "C:\Users\jhon\AppData\Roaming\Python\Python313\site-
packages\optuna\study\_optimize.py", line 201, in _run_trial
    value_or_values = func(trial)
  File "C:\Users\jhon\AppData\Local\Temp\ipykernel_21388\1645047760.py", line
67, in objective
    score = cross_val_score(model, X_train, y_train, cv=10,
scoring="accuracy").mean()
            ~~~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\utils\_param_validation.py", line 218, in wrapper
```

```
    return func(*args, **kwargs)
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\model_selection\_validation.py", line 677, in cross_val_score
    cv_results = cross_validate(
        estimator=estimator,
    …<9 lines>…
        error_score=error_score,
    )
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\utils\_param_validation.py", line 218, in wrapper
    return func(*args, **kwargs)
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\model_selection\_validation.py", line 399, in cross_validate
    results = parallel(
        delayed(_fit_and_score)(
    …<15 lines>…
        for train, test in indices
    )
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\utils\parallel.py", line 82, in __call__
    return super().__call__(iterable_with_config_and_warning_filters)
           ~~~~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\joblib\parallel.py", line 1986, in __call__
    return output if self.return_generator else list(output)
                                                 ~~~~^^^^^^^^
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\joblib\parallel.py", line 1914, in _get_sequential_output
    res = func(*args, **kwargs)
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\utils\parallel.py", line 147, in __call__
    return self.function(*args, **kwargs)
           ~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\model_selection\_validation.py", line 881, in _fit_and_score
    test_scores = _score(
        estimator, X_test, y_test, scorer, score_params_test, error_score
    )
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\model_selection\_validation.py", line 942, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\metrics\_scorer.py", line 152, in __call__
    score = scorer._score(
        cached_call, estimator, *args, **routed_params.get(name).score
    )
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\metrics\_scorer.py", line 400, in _score
```

```
    y_pred = method_caller(
        estimator,
    …<2 lines>…
        pos_label=pos_label,
    )
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\metrics\_scorer.py", line 90, in _cached_call
    result, _ = _get_response_values(
                ~~~~~~~~~~~~~~~~~~~~^
        estimator, *args, response_method=response_method, **kwargs
        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    )
    ^
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\utils\_response.py", line 214, in _get_response_values
    y_pred = prediction_method(X)
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\naive_bayes.py", line 106, in predict
    jll = self._joint_log_likelihood(X)
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\sklearn\naive_bayes.py", line 514, in _joint_log_likelihood
    n_ij -= 0.5 * np.sum(((X - self.theta_[i, :]) ** 2) / (self.var_[i, :]), 1)
                  ~~~~~~^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  File "c:\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-
packages\numpy\_core\fromnumeric.py", line 2333, in _sum_dispatcher
    def _sum_dispatcher(a, axis=None, dtype=None, out=None, keepdims=None,

KeyboardInterrupt
[W 2025-12-02 14:15:02,051] Trial 36 failed with value None.
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[243], line 72
     70 # Create and run the Optuna study
     71 study = optuna.create_study(direction="maximize", sampler=optuna.
 ↪samplers.TPESampler(seed=42))
---> 72 study.optimize(objective, n_trials=500, show_progress_bar=True)
     74 # Get the best model and hyperparameters
     75 best_params = study.best_params

File ~\AppData\Roaming\Python\Python313\site-packages\optuna\study\study.py:490
 ↪in Study.optimize(self, func, n_trials, timeout, n_jobs, catch, callbacks,␣
 ↪gc_after_trial, show_progress_bar)
    388 def optimize(
    389     self,
    390     func: ObjectiveFuncType,
   (…)    397     show_progress_bar: bool = False,
    398 ) -> None:
```

```
    399     """Optimize an objective function.
    400
    401     Optimization is done by choosing a suitable set of hyperparameter␣
 ↪values from a given
   (…)    488             If nested invocation of this method occurs.
    489     """
--> 490     _optimize(
    491         study=self,
    492         func=func,
    493         n_trials=n_trials,
    494         timeout=timeout,
    495         n_jobs=n_jobs,
    496         catch=tuple(catch) if isinstance(catch, Iterable) else (catch,)
    497         callbacks=callbacks,
    498         gc_after_trial=gc_after_trial,
    499         show_progress_bar=show_progress_bar,
    500     )

File ~\AppData\Roaming\Python\Python313\site-packages\optuna\study\_optimize.py
 ↪63, in _optimize(study, func, n_trials, timeout, n_jobs, catch, callbacks,␣
 ↪gc_after_trial, show_progress_bar)
     61 try:
     62     if n_jobs == 1:
---> 63         _optimize_sequential(
     64             study,
     65             func,
     66             n_trials,
     67             timeout,
     68             catch,
     69             callbacks,
     70             gc_after_trial,
     71             reseed_sampler_rng=False,
     72             time_start=None,
     73             progress_bar=progress_bar,
     74         )
     75     else:
     76         if n_jobs == -1:

File ~\AppData\Roaming\Python\Python313\site-packages\optuna\study\_optimize.py
 ↪160, in _optimize_sequential(study, func, n_trials, timeout, catch, callbacks␣
 ↪gc_after_trial, reseed_sampler_rng, time_start, progress_bar)
    157         break
    159 try:
--> 160     frozen_trial_id = _run_trial(study, func, catch)
    161 finally:
    162     # The following line mitigates memory problems that can be occurred␣
 ↪in some
```

```
    163     # environments (e.g., services that use computing containers such a␣
 ↪GitHub Actions).
    164     # Please refer to the following PR for further details:
    165     # https://github.com/optuna/optuna/pull/325.
    166     if gc_after_trial:

File ~\AppData\Roaming\Python\Python313\site-packages\optuna\study\_optimize.py
 ↪258, in _run_trial(study, func, catch)
    251         assert False, "Should not reach."
    253 if (
    254     updated_state == TrialState.FAIL
    255     and func_err is not None
    256     and not isinstance(func_err, catch)
    257 ):
--> 258     raise func_err
    259 return trial._trial_id

File ~\AppData\Roaming\Python\Python313\site-packages\optuna\study\_optimize.py
 ↪201, in _run_trial(study, func, catch)
    199 with get_heartbeat_thread(trial._trial_id, study._storage):
    200     try:
--> 201         value_or_values = func(trial)
    202     except exceptions.TrialPruned as e:
    203         # TODO(mamu): Handle multi-objective cases.
    204         state = TrialState.PRUNED

Cell In[243], line 67, in objective(trial)
     64     model = KNeighborsClassifier(**params)
     66 # Perform 10-fold cross-validation
---> 67 score =␣
 ↪cross_val_score(model, X_train, y_train, cv=10, scoring=          ).mean()
     68 return score

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\_param
 ↪py:218, in validate_params.<locals>.decorator.<locals>.wrapper(*args, **kwarg
    212 try:
    213     with config_context(
    214         skip_parameter_validation=(
    215             prefer_skip_nested_validation or global_skip_validation
    216         )
    217     ):
--> 218         return func(*args, **kwargs)
    219 except InvalidParameterError as e:
    220     # When the function is just a wrapper around an estimator, we allow
    221     # the function to delegate validation to the estimator, but we␣
 ↪replace
    222     # the name of the estimator by the name of the function in the erro
```

```
    223     # message to avoid confusion.
    224     msg = re.sub(
    225         r"parameter of \w+ must be",
    226         f"parameter of {func.__qualname__} must be",
    227         str(e),
    228     )

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\model_selecti
 ↪py:677, in cross_val_score(estimator, X, y, groups, scoring, cv, n_jobs,␣
 ↪verbose, params, pre_dispatch, error_score)
    674 # To ensure multimetric format is not supported
    675 scorer = check_scoring(estimator, scoring=scoring)
--> 677 cv_results = cross_validate(
    678     estimator=estimator,
    679     X=X,
    680     y=y,
    681     groups=groups,
    682     scoring={       : scorer},
    683     cv=cv,
    684     n_jobs=n_jobs,
    685     verbose=verbose,
    686     params=params,
    687     pre_dispatch=pre_dispatch,
    688     error_score=error_score,
    689 )
    690 return cv_results["test_score"]

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\_param
 ↪py:218, in validate_params.<locals>.decorator.<locals>.wrapper(*args, **kwarg)
    212 try:
    213     with config_context(
    214         skip_parameter_validation=(
    215             prefer_skip_nested_validation or global_skip_validation
    216         )
    217     ):
--> 218         return func(*args, **kwargs)
    219 except InvalidParameterError as e:
    220     # When the function is just a wrapper around an estimator, we allow
    221     # the function to delegate validation to the estimator, but we␣
 ↪replace
    222     # the name of the estimator by the name of the function in the erro
    223     # message to avoid confusion.
    224     msg = re.sub(
    225         r"parameter of \w+ must be",
    226         f"parameter of {func.__qualname__} must be",
    227         str(e),
```

```
    228     )

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\model_selecti
↪py:399, in cross_validate(estimator, X, y, groups, scoring, cv, n_jobs,␣
↪verbose, params, pre_dispatch, return_train_score, return_estimator,␣
↪return_indices, error_score)
    396 # We clone the estimator to make sure that all the folds are
    397 # independent, and that it is pickle-able.
    398 parallel = Parallel(n_jobs=n_jobs, verbose=verbose,␣
↪pre_dispatch=pre_dispatch)
--> 399 results = parallel(
    400     delayed(_fit_and_score)(
    401         clone(estimator),
    402         X,
    403         y,
    404         scorer=scorers,
    405         train=train,
    406         test=test,
    407         verbose=verbose,
    408         parameters=None,
    409         fit_params=routed_params.estimator.fit,
    410         score_params=routed_params.scorer.score,
    411         return_train_score=return_train_score,
    412         return_times=True,
    413         return_estimator=return_estimator,
    414         error_score=error_score,
    415     )
    416     for train, test in indices
    417 )
    419 _warn_or_raise_about_fit_failures(results, error_score)
    421 # For callable scoring, the return type is only know after calling. If␣
↪the
    422 # return type is a dictionary, the error scores can now be inserted wit
    423 # the correct key.

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\paralle
↪py:82, in Parallel.__call__(self, iterable)
     73 warning_filters = warnings.filters
     74 iterable_with_config_and_warning_filters = (
     75     (
     76         _with_config_and_warning_filters(delayed_func, config,␣
↪warning_filters),
   (…)     80     for delayed_func, args, kwargs in iterable
     81 )
---> 82 return super().__call__(iterable_with_config_and_warning_filters)
```

```
File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\joblib\parallel.
↪py:1986, in Parallel.__call__(self, iterable)
   1984     output = self._get_sequential_output(iterable)
   1985     next(output)
-> 1986     return output if self.return_generator else list(output)
   1988 # Let's create an ID that uniquely identifies the current call. If the
   1989 # call is interrupted early and that the same instance is immediately
   1990 # reused, this id will be used to prevent workers that were
   1991 # concurrently finalizing a task from the previous call to run the
   1992 # callback.
   1993 with self._lock:

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\joblib\parallel.
↪py:1914, in Parallel._get_sequential_output(self, iterable)
   1912 self.n_dispatched_batches += 1
   1913 self.n_dispatched_tasks += 1
-> 1914 res = func(*args, **kwargs)
   1915 self.n_completed_tasks += 1
   1916 self.print_progress()

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\parall
↪py:147, in _FuncWrapper.__call__(self, *args, **kwargs)
    145 with config_context(**config), warnings.catch_warnings():
    146     warnings.filters = warning_filters
--> 147     return self.function(*args, **kwargs)

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\model_selecti
↪py:881, in _fit_and_score(estimator, X, y, scorer, train, test, verbose,␣
↪parameters, fit_params, score_params, return_train_score, return_parameters,␣
↪return_n_test_samples, return_times, return_estimator, split_progress,␣
↪candidate_progress, error_score)
    878 result["fit_error"] = None
    880 fit_time = time.time() - start_time
--> 881 test_scores = _score(
    882     estimator, X_test, y_test, scorer, score_params_test, error_score
    883 )
    884 score_time = time.time() - start_time - fit_time
    885 if return_train_score:

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\model_selecti
↪py:942, in _score(estimator, X_test, y_test, scorer, score_params, error_score)
    940         scores = scorer(estimator, X_test, **score_params)
    941     else:
--> 942         scores = scorer(estimator, X_test, y_test, **score_params)
    943 except Exception:
```

```
    944     if isinstance(scorer, _MultimetricScorer):
    945         # If `_MultimetricScorer` raises exception, the `error_score`
    946         # parameter is equal to "raise".

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\metrics\_scor
↪py:152, in _MultimetricScorer.__call__(self, estimator, *args, **kwargs)
    150 try:
    151     if isinstance(scorer, _BaseScorer):
--> 152         score = scorer._score(
    153␣
↪            cached_call, estimator, *args, **routed_params.get(name).score
    154         )
    155     else:
    156         score = scorer(estimator, *args, **routed_params.get(name).scor )

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\metrics\_scor
↪py:400, in _Scorer._score(self, method_caller, estimator, X, y_true, **kwargs
    398 pos_label = None if is_regressor(estimator) else self._get_pos_label()
    399 response_method = _check_response_method(estimator, self.
↪_response_method)
--> 400 y_pred = method_caller(
    401     estimator,
    402     _get_response_method_name(response_method),
    403     X,
    404     pos_label=pos_label,
    405 )
    407 scoring_kwargs = {**self._kwargs, **kwargs}
    408 return self._sign * self._score_func(y_true, y_pred, **scoring_kwargs)

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\metrics\_scor
↪py:90, in _cached_call(cache, estimator, response_method, *args, **kwargs)
     87 if cache is not None and response_method in cache:
     88     return cache[response_method]
---> 90 result, _ = _get_response_values(
     91     estimator, *args, response_method=response_method, **kwargs
     92 )
     94 if cache is not None:
     95     cache[response_method] = result

File c:
↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\_respon
↪py:214, in _get_response_values(estimator, X, response_method, pos_label,␣
↪return_response_method_used)
    211     elif pos_label is None and target_type == "binary":
    212         pos_label = classes[-1]
--> 214 y_pred = prediction_method(X)
```

```
    216 if prediction_method.__name__ in ("predict_proba", "predict_log_proba")
    217     y_pred = _process_predict_proba(
    218         y_pred=y_pred,
    219         target_type=target_type,
    220         classes=classes,
    221         pos_label=pos_label,
    222     )

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\naive_bayes.
 ↪py:106, in _BaseNB.predict(self, X)
    104 check_is_fitted(self)
    105 X = self._check_X(X)
--> 106 jll = self._joint_log_likelihood(X)
    107 return self.classes_[np.argmax(jll, axis=1)]

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\naive_bayes.
 ↪py:514, in GaussianNB._joint_log_likelihood(self, X)
    512     jointi = np.log(self.class_prior_[i])
    513     n_ij = -0.5 * np.sum(np.log(2.0 * np.pi * self.var_[i, :]))
--> 514     n_ij -= 0.5 *␣
 ↪np.sum(((X - self.theta_[i, :]) ** 2) / (self.var_[i, :]), 1)
    515     joint_log_likelihood.append(jointi + n_ij)
    517 joint_log_likelihood = np.array(joint_log_likelihood).T

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\numpy\_core\fromnume
 ↪py:2333, in _sum_dispatcher(a, axis, dtype, out, keepdims, initial, where)
   2327         raise ValueError("Passing `min` or `max` keyword argument when
   2328                          "`a_min` and `a_max` are provided is forbidden
 ↪")
   2330     return _wrapfunc(a, 'clip', a_min, a_max, out=out, **kwargs)
-> 2333 def _sum_dispatcher(a, axis=None, dtype=None, out=None, keepdims=None,
   2334                     initial=None, where=None):
   2335     return (a, out)
   2338 @array_function_dispatch(_sum_dispatcher)
   2339 def sum(a, axis=None, dtype=None, out=None, keepdims=np._NoValue,
   2340         initial=np._NoValue, where=np._NoValue):

KeyboardInterrupt:
```

Item 2.b. What is the Accuracy and F1-score on the Test Data of the best model?

The best classifier is RandomForest (n_estimators=50, max_depth=17, min_samples_split=2), havning Test Accuracy of 98% and Test F1-Score of 98%.

1.2.4 2.IV. Random Forest

This section focuses on the following; finding a better Random Forest through hyperparameter tuning, and comparison on paper's Best Random Forest.

Item 2.c. In the paper, the best model was found to be Random Forest, having a weighted average F1 score of 0.98. Using your own hyper-parameter search, can you find a better Random Forest model with higher F1 score?

[ ]:
```
def objective_rf(trial):

    params = {
        "n_estimators": trial.suggest_int("n_estimators", 10, 500, step=2),
        "max_depth": trial.suggest_int("max_depth", 3, 50),
        "min_samples_split": trial.suggest_int("min_samples_split", 2, 10),
    }
    model = RandomForestClassifier(**params, random_state=42)

    # Perform 10-fold cross-validation
    score = cross_val_score(model, X_train, y_train, cv=10, scoring="accuracy").
 ↪mean()
    return score

# Create and run the Optuna study
study_rf = optuna.create_study(direction="maximize", sampler=optuna.samplers.
 ↪TPESampler(seed=42))
study_rf.optimize(objective_rf, n_trials=500, show_progress_bar=True)

# Get the best model and hyperparameters
best_params_rf = study_rf.best_params



best_model = RandomForestClassifier(**best_params_rf, random_state=42)

# Train the best model on the training data
best_model.fit(X_train, y_train)

# Evaluate on the test data
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average="weighted")

print(f"Best Hyperparameters: {best_params}")
print(f"Test Accuracy: {accuracy}")
print(f"Test F1-Score: {f1}")
```

```
[I 2025-12-02 13:54:45,466] A new study created in memory with name: no-
name-4ee2fda4-e265-4674-9c5c-00092cfd7811
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
[I 2025-12-02 13:54:46,823] Trial 0 finished with value: 0.9639372822299652 and
parameters: {'n_estimators': 194, 'max_depth': 48, 'min_samples_split': 8}. Best
is trial 0 with value: 0.9639372822299652.
[I 2025-12-02 13:54:48,916] Trial 1 finished with value: 0.9783391405342625 and
parameters: {'n_estimators': 304, 'max_depth': 10, 'min_samples_split': 3}. Best
is trial 1 with value: 0.9783391405342625.
[I 2025-12-02 13:54:49,207] Trial 2 finished with value: 0.9663182346109174 and
parameters: {'n_estimators': 38, 'max_depth': 44, 'min_samples_split': 7}. Best
is trial 1 with value: 0.9783391405342625.
[I 2025-12-02 13:54:51,492] Trial 3 finished with value: 0.9085946573751451 and
parameters: {'n_estimators': 358, 'max_depth': 3, 'min_samples_split': 10}. Best
is trial 1 with value: 0.9783391405342625.
[I 2025-12-02 13:54:54,357] Trial 4 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 418, 'max_depth': 13, 'min_samples_split': 3}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:54:55,063] Trial 5 finished with value: 0.9736353077816492 and
parameters: {'n_estimators': 100, 'max_depth': 17, 'min_samples_split': 6}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:54:56,583] Trial 6 finished with value: 0.9711962833914054 and
parameters: {'n_estimators': 222, 'max_depth': 16, 'min_samples_split': 7}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:54:57,135] Trial 7 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 78, 'max_depth': 17, 'min_samples_split': 5}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:54:58,728] Trial 8 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 234, 'max_depth': 40, 'min_samples_split': 3}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:55:00,510] Trial 9 finished with value: 0.9735772357723578 and
parameters: {'n_estimators': 262, 'max_depth': 31, 'min_samples_split': 2}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:55:03,773] Trial 10 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 490, 'max_depth': 29, 'min_samples_split': 4}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:55:06,680] Trial 11 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 438, 'max_depth': 20, 'min_samples_split': 5}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:55:07,596] Trial 12 finished with value: 0.966376306620209 and
parameters: {'n_estimators': 134, 'max_depth': 7, 'min_samples_split': 5}. Best
is trial 4 with value: 0.9783972125435539.
[I 2025-12-02 13:55:10,180] Trial 13 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 380, 'max_depth': 24, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:12,695] Trial 14 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 376, 'max_depth': 24, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:15,116] Trial 15 finished with value: 0.9783972125435539 and
```

```
parameters: {'n_estimators': 362, 'max_depth': 25, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:17,377] Trial 16 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 334, 'max_depth': 35, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:20,155] Trial 17 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 414, 'max_depth': 25, 'min_samples_split': 4}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:23,363] Trial 18 finished with value: 0.9590592334494772 and
parameters: {'n_estimators': 490, 'max_depth': 36, 'min_samples_split': 9}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:25,291] Trial 19 finished with value: 0.9736353077816492 and
parameters: {'n_estimators': 282, 'max_depth': 23, 'min_samples_split': 4}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:27,996] Trial 20 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 396, 'max_depth': 31, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:30,645] Trial 21 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 390, 'max_depth': 32, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:33,637] Trial 22 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 446, 'max_depth': 22, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:35,834] Trial 23 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 320, 'max_depth': 29, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:38,370] Trial 24 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 372, 'max_depth': 37, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:41,406] Trial 25 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 454, 'max_depth': 28, 'min_samples_split': 4}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:44,035] Trial 26 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 390, 'max_depth': 20, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:46,329] Trial 27 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 338, 'max_depth': 33, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:47,471] Trial 28 finished with value: 0.9736353077816492 and
parameters: {'n_estimators': 166, 'max_depth': 42, 'min_samples_split': 6}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:49,493] Trial 29 finished with value: 0.9663182346109174 and
parameters: {'n_estimators': 302, 'max_depth': 26, 'min_samples_split': 8}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:52,226] Trial 30 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 408, 'max_depth': 48, 'min_samples_split': 4}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:54,813] Trial 31 finished with value: 0.9807781649245063 and
```

```
parameters: {'n_estimators': 382, 'max_depth': 34, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:55:57,877] Trial 32 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 454, 'max_depth': 30, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:00,172] Trial 33 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 344, 'max_depth': 38, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:02,245] Trial 34 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 306, 'max_depth': 33, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:04,944] Trial 35 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 398, 'max_depth': 22, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:08,133] Trial 36 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 468, 'max_depth': 13, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:11,011] Trial 37 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 424, 'max_depth': 46, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:13,576] Trial 38 finished with value: 0.9711962833914054 and
parameters: {'n_estimators': 380, 'max_depth': 32, 'min_samples_split': 7}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:15,531] Trial 39 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 274, 'max_depth': 27, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:17,102] Trial 40 finished with value: 0.954239256678281 and
parameters: {'n_estimators': 228, 'max_depth': 19, 'min_samples_split': 10}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:19,732] Trial 41 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 392, 'max_depth': 19, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:22,096] Trial 42 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 356, 'max_depth': 24, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:24,998] Trial 43 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 434, 'max_depth': 12, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:28,161] Trial 44 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 474, 'max_depth': 21, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:30,942] Trial 45 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 398, 'max_depth': 18, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:31,267] Trial 46 finished with value: 0.9712543554006968 and
parameters: {'n_estimators': 40, 'max_depth': 9, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:33,891] Trial 47 finished with value: 0.9783972125435539 and
```

```
parameters: {'n_estimators': 362, 'max_depth': 16, 'min_samples_split': 5}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:36,802] Trial 48 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 424, 'max_depth': 27, 'min_samples_split': 4}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:39,129] Trial 49 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 324, 'max_depth': 31, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:42,626] Trial 50 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 500, 'max_depth': 39, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:45,104] Trial 51 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 376, 'max_depth': 35, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:47,713] Trial 52 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 386, 'max_depth': 41, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:50,543] Trial 53 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 412, 'max_depth': 24, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:52,852] Trial 54 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 346, 'max_depth': 34, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:54,912] Trial 55 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 304, 'max_depth': 30, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:57,348] Trial 56 finished with value: 0.9590011614401858 and
parameters: {'n_estimators': 366, 'max_depth': 28, 'min_samples_split': 9}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:56:58,676] Trial 57 finished with value: 0.9229965156794424 and
parameters: {'n_estimators': 206, 'max_depth': 4, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:00,461] Trial 58 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 254, 'max_depth': 16, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:02,733] Trial 59 finished with value: 0.9736353077816492 and
parameters: {'n_estimators': 326, 'max_depth': 25, 'min_samples_split': 4}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:05,772] Trial 60 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 440, 'max_depth': 21, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:08,451] Trial 61 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 392, 'max_depth': 19, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:11,245] Trial 62 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 406, 'max_depth': 15, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:13,874] Trial 63 finished with value: 0.9783972125435539 and
```

```
parameters: {'n_estimators': 382, 'max_depth': 23, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:16,286] Trial 64 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 352, 'max_depth': 29, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:19,519] Trial 65 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 466, 'max_depth': 36, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:22,347] Trial 66 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 420, 'max_depth': 32, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:25,056] Trial 67 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 396, 'max_depth': 26, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:27,537] Trial 68 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 372, 'max_depth': 21, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:30,455] Trial 69 finished with value: 0.9711962833914054 and
parameters: {'n_estimators': 434, 'max_depth': 18, 'min_samples_split': 7}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:32,715] Trial 70 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 332, 'max_depth': 23, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:35,436] Trial 71 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 398, 'max_depth': 18, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:38,182] Trial 72 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 408, 'max_depth': 20, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:40,601] Trial 73 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 358, 'max_depth': 14, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:43,241] Trial 74 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 388, 'max_depth': 17, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:46,191] Trial 75 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 452, 'max_depth': 19, 'min_samples_split': 6}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:49,072] Trial 76 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 428, 'max_depth': 34, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:49,975] Trial 77 finished with value: 0.9760162601626016 and
parameters: {'n_estimators': 132, 'max_depth': 28, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:51,926] Trial 78 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 290, 'max_depth': 31, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:54,042] Trial 79 finished with value: 0.9783972125435539 and
```

```
parameters: {'n_estimators': 314, 'max_depth': 11, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:56,368] Trial 80 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 344, 'max_depth': 25, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:57:58,909] Trial 81 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 374, 'max_depth': 36, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:01,478] Trial 82 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 372, 'max_depth': 38, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:04,240] Trial 83 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 398, 'max_depth': 33, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:06,807] Trial 84 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 382, 'max_depth': 35, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:09,582] Trial 85 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 414, 'max_depth': 30, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:12,079] Trial 86 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 360, 'max_depth': 32, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:14,376] Trial 87 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 340, 'max_depth': 24, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:17,082] Trial 88 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 402, 'max_depth': 22, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:20,188] Trial 89 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 446, 'max_depth': 37, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:23,035] Trial 90 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 422, 'max_depth': 42, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:25,606] Trial 91 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 380, 'max_depth': 41, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:28,129] Trial 92 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 366, 'max_depth': 43, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:31,172] Trial 93 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 386, 'max_depth': 50, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:33,835] Trial 94 finished with value: 0.9807781649245063 and
parameters: {'n_estimators': 392, 'max_depth': 27, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:36,183] Trial 95 finished with value: 0.9783972125435539 and
```

```
parameters: {'n_estimators': 350, 'max_depth': 46, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:39,046] Trial 96 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 406, 'max_depth': 34, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:41,865] Trial 97 finished with value: 0.9615563298490126 and
parameters: {'n_estimators': 414, 'max_depth': 40, 'min_samples_split': 8}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:44,858] Trial 98 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 430, 'max_depth': 20, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:47,485] Trial 99 finished with value: 0.9783972125435539 and
parameters: {'n_estimators': 372, 'max_depth': 45, 'min_samples_split': 3}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:49,763] Trial 100 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 332, 'max_depth': 22, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:52,428] Trial 101 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:55,102] Trial 102 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:58:57,541] Trial 103 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 364, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:00,310] Trial 104 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:02,934] Trial 105 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 380, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:06,070] Trial 106 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 462, 'max_depth': 14, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:09,034] Trial 107 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 440, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:11,413] Trial 108 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 354, 'max_depth': 29, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:14,180] Trial 109 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 416, 'max_depth': 21, 'min_samples_split': 5}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:17,345] Trial 110 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 478, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:19,973] Trial 111 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 398, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:22,597] Trial 112 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:25,122] Trial 113 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:27,570] Trial 114 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 374, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:30,431] Trial 115 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 422, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:33,143] Trial 116 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 408, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:35,518] Trial 117 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 362, 'max_depth': 23, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:38,157] Trial 118 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:39,322] Trial 119 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 170, 'max_depth': 39, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:41,813] Trial 120 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 376, 'max_depth': 18, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:44,480] Trial 121 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 15, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:47,072] Trial 122 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:49,757] Trial 123 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 410, 'max_depth': 18, 'min_samples_split': 6}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:52,614] Trial 124 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 428, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:55,058] Trial 125 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 368, 'max_depth': 37, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 13:59:57,396] Trial 126 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 352, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:00,070] Trial 127 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 400, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:02,543] Trial 128 finished with value: 0.9566202090592334
and parameters: {'n_estimators': 382, 'max_depth': 22, 'min_samples_split': 10}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:05,339] Trial 129 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 420, 'max_depth': 29, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:08,266] Trial 130 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 440, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:10,859] Trial 131 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:13,530] Trial 132 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 404, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:16,131] Trial 133 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:16,610] Trial 134 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 64, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:19,330] Trial 135 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:21,665] Trial 136 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 342, 'max_depth': 30, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:24,474] Trial 137 finished with value: 0.956678281068525 and
parameters: {'n_estimators': 414, 'max_depth': 15, 'min_samples_split': 9}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:26,950] Trial 138 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 362, 'max_depth': 36, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:29,608] Trial 139 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:29,729] Trial 140 finished with value: 0.9736933797909408
and parameters: {'n_estimators': 12, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:32,369] Trial 141 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 380, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:34,904] Trial 142 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:37,654] Trial 143 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 402, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:39,408] Trial 144 finished with value: 0.9735772357723578
and parameters: {'n_estimators': 248, 'max_depth': 38, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:42,066] Trial 145 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 36, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:44,884] Trial 146 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 410, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:47,311] Trial 147 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 354, 'max_depth': 34, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:50,008] Trial 148 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:52,963] Trial 149 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 430, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:55,555] Trial 150 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 376, 'max_depth': 26, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:00:58,175] Trial 151 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 42, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:00,683] Trial 152 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 37, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:03,423] Trial 153 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 398, 'max_depth': 41, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:06,021] Trial 154 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 380, 'max_depth': 44, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:08,522] Trial 155 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 364, 'max_depth': 30, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:11,398] Trial 156 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 418, 'max_depth': 39, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:14,192] Trial 157 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 406, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:16,862] Trial 158 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 41, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:19,329] Trial 159 finished with value: 0.9783972125435539
```

```
and parameters: {'n_estimators': 358, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:21,915] Trial 160 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 380, 'max_depth': 25, 'min_samples_split': 6}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:24,609] Trial 161 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 50, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:27,376] Trial 162 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 47, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:29,982] Trial 163 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 50, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:32,801] Trial 164 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 414, 'max_depth': 49, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:35,347] Trial 165 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:38,035] Trial 166 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:40,393] Trial 167 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 346, 'max_depth': 47, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:42,973] Trial 168 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 44, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:45,735] Trial 169 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 406, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:48,371] Trial 170 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:51,072] Trial 171 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:53,577] Trial 172 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 368, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:56,232] Trial 173 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:01:59,096] Trial 174 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 420, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:01,793] Trial 175 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 400, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:04,373] Trial 176 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:07,210] Trial 177 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 412, 'max_depth': 21, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:09,696] Trial 178 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 362, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:12,368] Trial 179 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:15,134] Trial 180 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 398, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:17,740] Trial 181 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 382, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:20,405] Trial 182 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:22,938] Trial 183 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 372, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:25,708] Trial 184 finished with value: 0.9711962833914054
and parameters: {'n_estimators': 408, 'max_depth': 23, 'min_samples_split': 7}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:28,323] Trial 185 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:31,054] Trial 186 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 398, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:33,937] Trial 187 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 424, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:36,352] Trial 188 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 354, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:38,905] Trial 189 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 376, 'max_depth': 30, 'min_samples_split': 5}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:41,695] Trial 190 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 406, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:44,388] Trial 191 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 396, 'max_depth': 14, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:47,078] Trial 192 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:49,757] Trial 193 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:52,286] Trial 194 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 366, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:54,885] Trial 195 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 374, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:02:57,682] Trial 196 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 408, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:00,318] Trial 197 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 382, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:03,044] Trial 198 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 15, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:05,919] Trial 199 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 416, 'max_depth': 36, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:08,676] Trial 200 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:11,509] Trial 201 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 40, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:14,190] Trial 202 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 380, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:17,035] Trial 203 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 404, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:19,767] Trial 204 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:22,293] Trial 205 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 364, 'max_depth': 29, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:24,855] Trial 206 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:27,649] Trial 207 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 400, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:30,522] Trial 208 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 414, 'max_depth': 38, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:31,374] Trial 209 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 118, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:34,141] Trial 210 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:36,809] Trial 211 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:39,346] Trial 212 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:41,949] Trial 213 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 382, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:44,710] Trial 214 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 398, 'max_depth': 13, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:47,297] Trial 215 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 43, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:49,918] Trial 216 finished with value: 0.9615563298490126
and parameters: {'n_estimators': 390, 'max_depth': 33, 'min_samples_split': 8}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:52,446] Trial 217 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 362, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:55,294] Trial 218 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 408, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:03:58,223] Trial 219 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 428, 'max_depth': 15, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:01,036] Trial 220 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 398, 'max_depth': 18, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:03,669] Trial 221 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 45, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:06,391] Trial 222 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:09,233] Trial 223 finished with value: 0.9783972125435539
```

```
and parameters: {'n_estimators': 406, 'max_depth': 29, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:11,786] Trial 224 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 372, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:14,483] Trial 225 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:16,013] Trial 226 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 218, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:18,870] Trial 227 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 418, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:21,422] Trial 228 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 30, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:24,153] Trial 229 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:26,905] Trial 230 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 404, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:29,511] Trial 231 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:32,304] Trial 232 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:34,907] Trial 233 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 382, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:36,729] Trial 234 finished with value: 0.9735772357723578
and parameters: {'n_estimators': 270, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:39,518] Trial 235 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 410, 'max_depth': 48, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:42,235] Trial 236 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:44,772] Trial 237 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 374, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:47,555] Trial 238 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:50,197] Trial 239 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 384, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:52,719] Trial 240 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:55,494] Trial 241 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:04:58,096] Trial 242 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 382, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:00,792] Trial 243 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:03,564] Trial 244 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:06,154] Trial 245 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 36, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:08,951] Trial 246 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 410, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:11,677] Trial 247 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 398, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:14,118] Trial 248 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 362, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:16,754] Trial 249 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:19,353] Trial 250 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:22,072] Trial 251 finished with value: 0.9759581881533101
and parameters: {'n_estimators': 392, 'max_depth': 9, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:24,778] Trial 252 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:27,550] Trial 253 finished with value: 0.956678281068525 and
parameters: {'n_estimators': 416, 'max_depth': 23, 'min_samples_split': 9}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:30,138] Trial 254 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:32,834] Trial 255 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 392, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:33,983] Trial 256 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 162, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:36,845] Trial 257 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 406, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:39,413] Trial 258 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:42,024] Trial 259 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:44,539] Trial 260 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 358, 'max_depth': 37, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:47,252] Trial 261 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 41, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:50,189] Trial 262 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 422, 'max_depth': 30, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:52,791] Trial 263 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 378, 'max_depth': 32, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:55,609] Trial 264 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:05:58,351] Trial 265 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:01,004] Trial 266 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:03,742] Trial 267 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 410, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:06,245] Trial 268 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 368, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:08,872] Trial 269 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:11,365] Trial 270 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:13,760] Trial 271 finished with value: 0.9760162601626016
```

```
and parameters: {'n_estimators': 352, 'max_depth': 20, 'min_samples_split': 4}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:16,604] Trial 272 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 416, 'max_depth': 29, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:19,350] Trial 273 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:21,986] Trial 274 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 45, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:24,618] Trial 275 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:27,424] Trial 276 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:29,962] Trial 277 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 366, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:32,779] Trial 278 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 408, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:35,490] Trial 279 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 382, 'max_depth': 43, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:38,238] Trial 280 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:40,996] Trial 281 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:43,570] Trial 282 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 374, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:46,233] Trial 283 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 40, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:49,199] Trial 284 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 432, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:52,032] Trial 285 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 412, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:54,787] Trial 286 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 15, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:57,357] Trial 287 finished with value: 0.9711962833914054
```

```
and parameters: {'n_estimators': 364, 'max_depth': 33, 'min_samples_split': 7}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:06:59,986] Trial 288 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 384, 'max_depth': 49, 'min_samples_split': 5}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:02,848] Trial 289 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 422, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:06,227] Trial 290 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 498, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:09,095] Trial 291 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 404, 'max_depth': 38, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:11,714] Trial 292 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 36, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:14,433] Trial 293 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 396, 'max_depth': 31, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:16,949] Trial 294 finished with value: 0.9085946573751451
and parameters: {'n_estimators': 386, 'max_depth': 3, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:19,394] Trial 295 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 354, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:21,906] Trial 296 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 46, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:24,663] Trial 297 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 398, 'max_depth': 24, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:27,507] Trial 298 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 410, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:30,174] Trial 299 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 382, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:32,850] Trial 300 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:35,390] Trial 301 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 376, 'max_depth': 17, 'min_samples_split': 4}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:38,251] Trial 302 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 14, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:40,918] Trial 303 finished with value: 0.9783972125435539
```

```
and parameters: {'n_estimators': 364, 'max_depth': 39, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:43,806] Trial 304 finished with value: 0.966376306620209 and
parameters: {'n_estimators': 416, 'max_depth': 6, 'min_samples_split': 2}. Best
is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:45,777] Trial 305 finished with value: 0.9566202090592334
and parameters: {'n_estimators': 286, 'max_depth': 29, 'min_samples_split': 10}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:46,381] Trial 306 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 80, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:49,116] Trial 307 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:51,738] Trial 308 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 380, 'max_depth': 37, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:54,505] Trial 309 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:57,333] Trial 310 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 408, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:07:59,734] Trial 311 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 344, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:02,336] Trial 312 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 372, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:05,111] Trial 313 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:07,636] Trial 314 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 358, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:10,485] Trial 315 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:13,573] Trial 316 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 424, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:15,879] Trial 317 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 312, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:18,673] Trial 318 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 394, 'max_depth': 30, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:21,368] Trial 319 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 380, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:24,238] Trial 320 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 410, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:26,845] Trial 321 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:29,858] Trial 322 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:32,542] Trial 323 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 388, 'max_depth': 42, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:33,886] Trial 324 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 186, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:36,538] Trial 325 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:39,273] Trial 326 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:42,142] Trial 327 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 418, 'max_depth': 30, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:44,985] Trial 328 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 406, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:47,568] Trial 329 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 366, 'max_depth': 47, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:50,257] Trial 330 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 384, 'max_depth': 22, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:53,369] Trial 331 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 438, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:55,475] Trial 332 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 298, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:08:58,319] Trial 333 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:01,031] Trial 334 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:03,651] Trial 335 finished with value: 0.9807781649245063
```

```
and parameters: {'n_estimators': 376, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:06,107] Trial 336 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 360, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:08,931] Trial 337 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 410, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:11,601] Trial 338 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 36, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:14,306] Trial 339 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 394, 'max_depth': 28, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:16,975] Trial 340 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 372, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:18,766] Trial 341 finished with value: 0.9735772357723578
and parameters: {'n_estimators': 248, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:21,514] Trial 342 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:24,186] Trial 343 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 29, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:27,061] Trial 344 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 412, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:29,778] Trial 345 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 44, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:32,739] Trial 346 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 426, 'max_depth': 15, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:36,039] Trial 347 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 482, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:38,615] Trial 348 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:41,390] Trial 349 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 404, 'max_depth': 26, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:43,921] Trial 350 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 364, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:46,263] Trial 351 finished with value: 0.9783972125435539
```

```
and parameters: {'n_estimators': 334, 'max_depth': 34, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:48,968] Trial 352 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:51,413] Trial 353 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 354, 'max_depth': 50, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:54,048] Trial 354 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 380, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:56,788] Trial 355 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 400, 'max_depth': 33, 'min_samples_split': 6}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:09:59,302] Trial 356 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:02,106] Trial 357 finished with value: 0.9615563298490126
and parameters: {'n_estimators': 418, 'max_depth': 41, 'min_samples_split': 8}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:04,791] Trial 358 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:07,569] Trial 359 finished with value: 0.9566202090592334
and parameters: {'n_estimators': 396, 'max_depth': 35, 'min_samples_split': 9}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:10,385] Trial 360 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 404, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:13,060] Trial 361 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 380, 'max_depth': 14, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:15,827] Trial 362 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 29, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:18,751] Trial 363 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 414, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:21,507] Trial 364 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:25,403] Trial 365 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 37, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:29,278] Trial 366 finished with value: 0.9711962833914054
and parameters: {'n_estimators': 370, 'max_depth': 22, 'min_samples_split': 7}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:33,304] Trial 367 finished with value: 0.9783972125435539
```

```
and parameters: {'n_estimators': 408, 'max_depth': 16, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:37,223] Trial 368 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:40,849] Trial 369 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 348, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:44,464] Trial 370 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 360, 'max_depth': 48, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:48,207] Trial 371 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:51,944] Trial 372 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 390, 'max_depth': 39, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:55,764] Trial 373 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 378, 'max_depth': 12, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:10:59,928] Trial 374 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 406, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:03,603] Trial 375 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 420, 'max_depth': 43, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:07,519] Trial 376 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:11,130] Trial 377 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 370, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:14,113] Trial 378 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 382, 'max_depth': 36, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:17,702] Trial 379 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 398, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:21,956] Trial 380 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 432, 'max_depth': 30, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:25,464] Trial 381 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:29,495] Trial 382 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 412, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:33,171] Trial 383 finished with value: 0.9783972125435539
```

```
and parameters: {'n_estimators': 366, 'max_depth': 34, 'min_samples_split': 5}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:37,564] Trial 384 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 452, 'max_depth': 28, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:41,137] Trial 385 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:44,937] Trial 386 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 392, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:49,245] Trial 387 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 404, 'max_depth': 46, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:52,745] Trial 388 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 384, 'max_depth': 26, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:11:56,928] Trial 389 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 396, 'max_depth': 32, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:00,677] Trial 390 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 372, 'max_depth': 35, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:04,456] Trial 391 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 358, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:08,761] Trial 392 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 412, 'max_depth': 17, 'min_samples_split': 3}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:12,495] Trial 393 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 386, 'max_depth': 24, 'min_samples_split': 6}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:16,614] Trial 394 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 402, 'max_depth': 38, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:19,733] Trial 395 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 378, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:22,594] Trial 396 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 31, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:27,069] Trial 397 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 424, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:30,909] Trial 398 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 27, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:33,820] Trial 399 finished with value: 0.9760162601626016
```

```
and parameters: {'n_estimators': 406, 'max_depth': 34, 'min_samples_split': 4}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:36,411] Trial 400 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 366, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:39,234] Trial 401 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 376, 'max_depth': 15, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:42,030] Trial 402 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 394, 'max_depth': 25, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:44,923] Trial 403 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 414, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:47,722] Trial 404 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 400, 'max_depth': 33, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:50,393] Trial 405 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 386, 'max_depth': 29, 'min_samples_split': 2}.
Best is trial 13 with value: 0.9807781649245063.
[I 2025-12-02 14:12:50,539] Trial 406 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 16, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:12:50,965] Trial 407 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 56, 'max_depth': 15, 'min_samples_split': 3}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:12:53,817] Trial 408 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 406, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:12:54,639] Trial 409 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 112, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:12:57,627] Trial 410 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 418, 'max_depth': 14, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:12:58,300] Trial 411 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 88, 'max_depth': 16, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:12:59,781] Trial 412 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 204, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:00,094] Trial 413 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:00,484] Trial 414 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 50, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:00,822] Trial 415 finished with value: 0.980836236933798 and
```

```
parameters: {'n_estimators': 42, 'max_depth': 18, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:02,074] Trial 416 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 170, 'max_depth': 18, 'min_samples_split': 3}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:03,218] Trial 417 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 160, 'max_depth': 18, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:03,534] Trial 418 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 40, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:03,950] Trial 419 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 54, 'max_depth': 17, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:04,073] Trial 420 finished with value: 0.9736933797909408
and parameters: {'n_estimators': 12, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:04,447] Trial 421 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 34, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:04,778] Trial 422 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 28, 'max_depth': 21, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:05,089] Trial 423 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 24, 'max_depth': 20, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:05,392] Trial 424 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 26, 'max_depth': 21, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:05,692] Trial 425 finished with value: 0.9735772357723578
and parameters: {'n_estimators': 24, 'max_depth': 21, 'min_samples_split': 3}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:06,062] Trial 426 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 32, 'max_depth': 21, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:06,504] Trial 427 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 32, 'max_depth': 21, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:06,957] Trial 428 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 40, 'max_depth': 19, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:07,423] Trial 429 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 36, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:07,729] Trial 430 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 34, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:08,004] Trial 431 finished with value: 0.9832171893147503
```

```
and parameters: {'n_estimators': 34, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:08,281] Trial 432 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 34, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:08,550] Trial 433 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 30, 'max_depth': 21, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:08,796] Trial 434 finished with value: 0.9664343786295007
and parameters: {'n_estimators': 28, 'max_depth': 21, 'min_samples_split': 8}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:09,104] Trial 435 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:09,428] Trial 436 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:09,728] Trial 437 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:10,033] Trial 438 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:10,358] Trial 439 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 40, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:10,831] Trial 440 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 40, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:11,270] Trial 441 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:11,728] Trial 442 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:12,128] Trial 443 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 40, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:12,462] Trial 444 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:12,979] Trial 445 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 40, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:13,413] Trial 446 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 42, 'max_depth': 22, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:13,857] Trial 447 finished with value: 0.980836236933798 and
```

```
parameters: {'n_estimators': 42, 'max_depth': 22, 'min_samples_split': 2}. Best
is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:14,199] Trial 448 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 406 with value: 0.9832171893147503.
[I 2025-12-02 14:13:14,400] Trial 449 finished with value: 0.9855981416957027
and parameters: {'n_estimators': 20, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:14,559] Trial 450 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 16, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:15,092] Trial 451 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 66, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:15,235] Trial 452 finished with value: 0.9784552845528456
and parameters: {'n_estimators': 14, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:15,608] Trial 453 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 44, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:15,957] Trial 454 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 44, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:16,317] Trial 455 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 46, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:16,750] Trial 456 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 58, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:17,043] Trial 457 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:17,229] Trial 458 finished with value: 0.9855981416957027
and parameters: {'n_estimators': 20, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:17,397] Trial 459 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 18, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:17,899] Trial 460 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 64, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:18,182] Trial 461 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 36, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:18,329] Trial 462 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 16, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:18,615] Trial 463 finished with value: 0.9832171893147503
```

```
and parameters: {'n_estimators': 36, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:19,022] Trial 464 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 52, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:19,308] Trial 465 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 36, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:19,490] Trial 466 finished with value: 0.9855981416957027
and parameters: {'n_estimators': 20, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:19,681] Trial 467 finished with value: 0.9855981416957027
and parameters: {'n_estimators': 20, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:19,877] Trial 468 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 22, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:20,092] Trial 469 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 24, 'max_depth': 21, 'min_samples_split': 2}. Best
is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:20,490] Trial 470 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 52, 'max_depth': 20, 'min_samples_split': 3}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:21,036] Trial 471 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 70, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:21,309] Trial 472 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 34, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:21,431] Trial 473 finished with value: 0.9784552845528456
and parameters: {'n_estimators': 10, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:22,035] Trial 474 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 46, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:22,453] Trial 475 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 34, 'max_depth': 21, 'min_samples_split': 3}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:22,698] Trial 476 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 22, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:23,238] Trial 477 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 58, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:23,618] Trial 478 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 42, 'max_depth': 22, 'min_samples_split': 2}. Best
is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:23,923] Trial 479 finished with value: 0.980836236933798 and
```

```
parameters: {'n_estimators': 28, 'max_depth': 20, 'min_samples_split': 2}. Best
is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:24,094] Trial 480 finished with value: 0.9784552845528456
and parameters: {'n_estimators': 10, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:24,597] Trial 481 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 50, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:25,034] Trial 482 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 34, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:25,275] Trial 483 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 22, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:25,760] Trial 484 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 40, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:26,079] Trial 485 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 30, 'max_depth': 22, 'min_samples_split': 2}. Best
is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:26,777] Trial 486 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 60, 'max_depth': 20, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:27,272] Trial 487 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 48, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:27,557] Trial 488 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 18, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:27,988] Trial 489 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:28,865] Trial 490 finished with value: 0.9783972125435539
and parameters: {'n_estimators': 76, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:29,182] Trial 491 finished with value: 0.9760162601626016
and parameters: {'n_estimators': 28, 'max_depth': 20, 'min_samples_split': 5}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:29,700] Trial 492 finished with value: 0.9807781649245063
and parameters: {'n_estimators': 50, 'max_depth': 24, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:29,945] Trial 493 finished with value: 0.9855981416957027
and parameters: {'n_estimators': 20, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:30,183] Trial 494 finished with value: 0.9712543554006968
and parameters: {'n_estimators': 22, 'max_depth': 21, 'min_samples_split': 3}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:30,394] Trial 495 finished with value: 0.9736353077816492
```

```
and parameters: {'n_estimators': 18, 'max_depth': 23, 'min_samples_split': 4}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:30,746] Trial 496 finished with value: 0.980836236933798 and
parameters: {'n_estimators': 30, 'max_depth': 20, 'min_samples_split': 2}. Best
is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:30,892] Trial 497 finished with value: 0.9736933797909408
and parameters: {'n_estimators': 12, 'max_depth': 22, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:31,465] Trial 498 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 48, 'max_depth': 21, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
[I 2025-12-02 14:13:31,862] Trial 499 finished with value: 0.9832171893147503
and parameters: {'n_estimators': 38, 'max_depth': 23, 'min_samples_split': 2}.
Best is trial 449 with value: 0.9855981416957027.
Best Hyperparameters: {'n_estimators': 50, 'max_depth': 17, 'min_samples_split':
2}
Test Accuracy: 0.9807692307692307
Test F1-Score: 0.9808511271925906
```

Item 2.c. Using your own hyper-parameter search, can you find a better Random Forest model with higher F1 score?

A better Random Foreset Model with higher F1 score has parameters of n_estimators=50, max_depth=17, and min_samples_split=2.

[ ]:
```
categorical_columns = X.select_dtypes(include=['object', 'category']).columns.
 ↪tolist()
categorical_columns
numerical_columns = X.select_dtypes(include=['int64', 'float64']).columns.
 ↪tolist()
numerical_columns
```

[ ]:
```
['child_mort',
 'exports',
 'health',
 'imports',
 'income',
 'inflation',
 'life_expec',
 'total_fer',
 'gdpp']
```

Item 2.d. Based on the best Random Forest model in item (c), perform any feature importance method to explain the model. What insights can we get from the model?

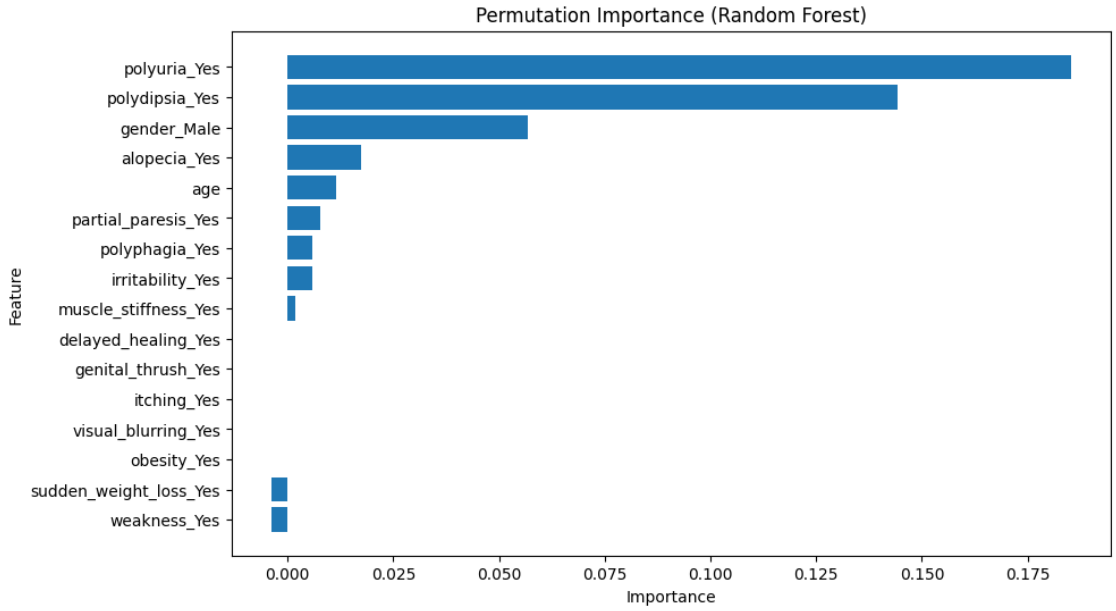
The results indicate that the top three features significantly influencing the model's performance are: polyuria_Yes, polydipsia_Yes, and gender_Male. These features play a critical role in the model's predictions and provide valuable insights into the factors driving its performance.

[ ]:
```
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OrdinalEncoder

categorical_columns = X.select_dtypes(include=['object', 'category']).columns.
 ↪tolist()
numerical_columns = X.select_dtypes(include=['int64', 'float64']).columns.
 ↪tolist()
categorical_encoder = OrdinalEncoder(
    handle_unknown="use_encoded_value", unknown_value=-1,␣
 ↪encoded_missing_value=-1
)
numerical_pipe = SimpleImputer(strategy="mean")

preprocessing = ColumnTransformer(
    [

        ("num", numerical_pipe, numerical_columns),
    ],
    verbose_feature_names_out=False,
)

rf = Pipeline(
    [
        ("preprocess", preprocessing),
        ("classifier", RandomForestClassifier(**best_params_rf,␣
 ↪random_state=42)),
    ]
)
rf.fit(X_train, y_train)
```

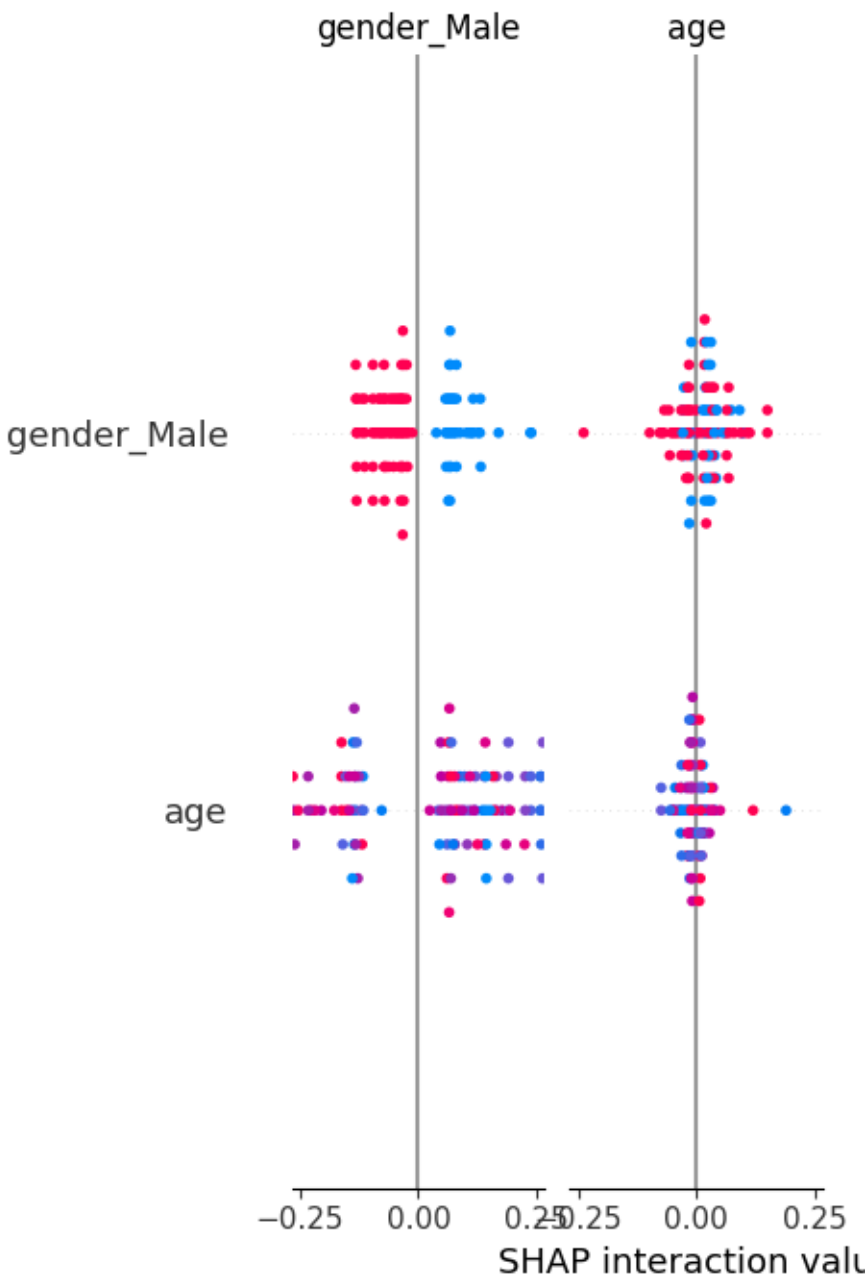
[ ]:
```
Pipeline(steps=[('preprocess',
                 ColumnTransformer(transformers=[('num', SimpleImputer(),
                                                  ['age'])],
                                   verbose_feature_names_out=False)),
                ('classifier',
                 RandomForestClassifier(max_depth=22, n_estimators=20,
                                        random_state=42))])
```

[ ]:
```
from sklearn.inspection import permutation_importance

# Calculate permutation importance
perm_importance = permutation_importance(best_model, X_test, y_test,␣
 ↪scoring="f1_weighted")

# Create a DataFrame for better visualization
```

```
perm_importance_df = pd.DataFrame({
    "Feature": X_train.columns,
    "Importance": perm_importance.importances_mean
}).sort_values(by="Importance", ascending=False)

# Plot the permutation importance
plt.figure(figsize=(10, 6))
plt.barh(perm_importance_df["Feature"], perm_importance_df["Importance"])
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.title("Permutation Importance (Random Forest)")
plt.gca().invert_yaxis()
plt.show()
```



[ ]:
```
import shap

# Create a SHAP explainer for the Random Forest model
explainer = shap.TreeExplainer(best_model)
shap_values = explainer.shap_values(X_test)

# Summary plot for global feature importance
shap.summary_plot(shap_values, X_test, plot_type="bar")

# Detailed plot for individual predictions
```



[ ]:
```
import shap

print(best_model)

# The next few lines help us apply SHAP values on a Pipeline model
explainer = shap.TreeExplainer(best_model)
observations = X_test
shap_values = explainer.shap_values(observations)
```

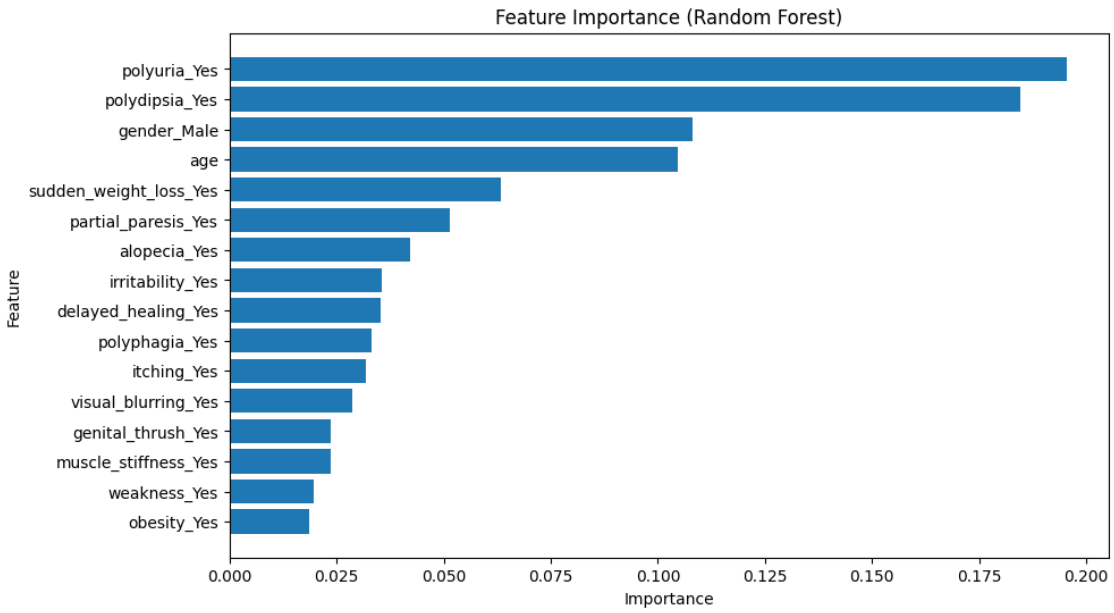
```
# Plot the feature importance using SHAP
shap.summary_plot(shap_values, X_test, plot_type="bar")
```

```
Pipeline(steps=[('scaler', StandardScaler()),
                ('ridge', Ridge(alpha=np.float64(46.41588833612773)))])
```

```
---------------------------------------------------------------------------
InvalidModelError                         Traceback (most recent call last)
Cell In[208], line 6
      3 print(best_model)
      5 # The next few lines help us apply SHAP values on a Pipeline model
----> 6 explainer = shap.TreeExplainer(best_model)
      7 observations = X_test
      8 shap_values = explainer.shap_values(observations)

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\shap\explainers\_tree
 ↪py:279, in TreeExplainer.__init__(self, model, data, model_output,␣
 ↪feature_perturbation, feature_names, approximate, link, linearize_link)
    277 self.feature_perturbation = feature_perturbation
    278 self.expected_value = None
--> 279 self.model =␣
 ↪TreeEnsemble(model, self.data, self.data_missing, model_output)
    280 self.model_output = model_output
    281 # self.model_output = self.model.model_output # this allows the␣
 ↪TreeEnsemble to translate model outputs types by how it loads the model
    282
    283 # check for unsupported combinations of feature_perturbation and␣
 ↪model_outputs

File c:
 ↪\Users\jhon\AppData\Local\Programs\Python\Python313\Lib\site-packages\shap\explainers\_tree
 ↪py:1443, in TreeEnsemble.__init__(self, model, data, data_missing,␣
 ↪model_output)
  1441     self.base_offset = model.init_params[param_idx]
  1442 else:
-> 1443     raise InvalidModelError("Model type not yet supported by␣
 ↪TreeExplainer: " + str(type(model)))
  1445 # build a dense numpy version of all the tree objects
  1446 if self.trees is not None and self.trees:

InvalidModelError: Model type not yet supported by TreeExplainer: <class␣
 ↪'sklearn.pipeline.Pipeline'>
```

[ ]:
```
import matplotlib.pyplot as plt
import pandas as pd
```

```
# Get feature importance from the trained Random Forest model
importances = best_model.feature_importances_

# Create a DataFrame for better visualization
feature_importance_df = pd.DataFrame({
    "Feature": X_train.columns,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)

# Plot the feature importance
plt.figure(figsize=(10, 6))
plt.barh(feature_importance_df["Feature"], feature_importance_df["Importance"])
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.title("Feature Importance (Random Forest)")
plt.gca().invert_yaxis()
plt.show()
```



[ ]:
```
from sklearn.pipeline import Pipeline
from sklearn.linear_model import Ridge
from sklearn.metrics import r2_score, accuracy_score

alphas = np.logspace(-3, 3, 10)
best_alpha = None
best_val_score = -np.inf
```

```
best_model = None

X_val = X_test
y_val = y_test
for alpha in alphas:
    pipe = Pipeline([
        ('scaler', StandardScaler()),
        ('ridge', Ridge(alpha=alpha))
    ])
    pipe.fit(X_train, y_train)
    val_pred = pipe.predict(X_val)
    val_score = r2_score(y_val, val_pred)
    if val_score > best_val_score:
        best_val_score = val_score
        best_alpha = alpha
        best_model = pipe

# Final evaluation
train_score =  r2_score(y_train, best_model.predict(X_train))
test_score = r2_score(y_test, best_model.predict(X_test))

ridge = best_model.named_steps['ridge']

# Get the feature names for X1 to X8
feature_names = np.array(X.columns[:8])

# Get absolute value of coefficients for the best Ridge model
coef_abs = np.abs(ridge.coef_[:8])

# Get indices of top 5 features
top5_idx = np.argsort(coef_abs)[-5:][::-1]
top5_features = feature_names[top5_idx]

print("Top 5 features among X1 to X8:", list(top5_features))
```

```
Top 5 features among X1 to X8: ['polyuria_Yes', 'polydipsia_Yes', 'gender_Male',
'genital_thrush_Yes', 'sudden_weight_loss_Yes']
```