In [1]:
```
import glob
import os

import pandas as pd
import numpy as np

from sklearn.metrics import f1_score
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.tree import DecisionTreeClassifier
# Random Forest

from sklearn.ensemble import RandomForestClassifier

from hyperopt import fmin, tpe, hp
from sklearn.model_selection import cross_val_score
from statistics import mean, stdev, median, mode
```

In [2]:
```
file_list = glob.glob(os.path.join(os.getcwd(), r"C:\Users\kalya\OneDrive - University of Illinois at Chicago\!UIC\!!Data Science\!Project\News Articles\bbc-fulltext\bbc\business/", "*.txt"))

#all_files = os.listdir(r"C:\Users\kalya\OneDrive - University of Illinois at Chicago\!UIC\Data Science\Project\News Articles\bbc-fulltext\bbc\business/")
# imagine you're one directory above test dir
corpus = []

for file_path in file_list:
    with open(file_path) as f_input:
        corpus.append(f_input.read())



data1 = pd.DataFrame(data = corpus, columns = ["Doc Name"])
#data = data.reset_index()
data1["Topic"] = "business"
data1.columns = ["Doc Text", "Topic"]
data1.head()
```

Out[2]:

	Doc Text	Topic
0	Ad sales boost Time Warner profit\n\nQuarterly...	business
1	Dollar gains on Greenspan speech\n\nThe dollar...	business
2	Yukos unit buyer faces loan claim\n\nThe owner...	business
3	High fuel prices hit BA's profits\n\nBritish A...	business
4	Pernod takeover talk lifts Domecq\n\nShares in...	business

In [3]:
```
file_list2 = glob.glob(os.path.join(os.getcwd(), r"C:\Users\kalya\OneDrive - U
niversity of Illinois at Chicago\!UIC\!!Data Science\!Project\News Articles\bb
c-fulltext\bbc\entertainment/", "*.txt"))

#all_files = os.listdir(r"C:\Users\kalya\OneDrive - University of Illinois at
 Chicago\!UIC\Data Science\Project\News Articles\bbc-fulltext\bbc\business/")
# imagine you're one directory above test dir
corpus2 = []
for file_path in file_list2:
    with open(file_path) as f_input:
        corpus2.append(f_input.read())

data2 = pd.DataFrame(data = corpus2, columns = ["Doc Name"])
#data = data.reset_index()
data2["Topic"] = "entertainment"
data2.columns = ["Doc Text", "Topic"]
data2.head()
```

Out[3]:

	Doc Text	Topic
0	Gallery unveils interactive tree\n\nA Christma...	entertainment
1	Jarre joins fairytale celebration\n\nFrench mu...	entertainment
2	Musical treatment for Capra film\n\nThe classi...	entertainment
3	Richard and Judy choose top books\n\nThe 10 au...	entertainment
4	Poppins musical gets flying start\n\nThe stage...	entertainment

In [4]:
```
file_list3 = glob.glob(os.path.join(os.getcwd(), r"C:\Users\kalya\OneDrive - University of Illinois at Chicago\!UIC\!!Data Science\!Project\News Articles\bbc-fulltext\bbc\politics/", "*.txt"))

#all_files = os.listdir(r"C:\Users\kalya\OneDrive - University of Illinois at Chicago\!UIC\Data Science\Project\News Articles\bbc-fulltext\bbc\business/")
# imagine you're one directory above test dir
corpus3 = []
for file_path in file_list3:
    with open(file_path) as f_input:
        corpus3.append(f_input.read())

data3 = pd.DataFrame(data = corpus3, columns = ["Doc Name"])
#data = data.reset_index()
data3["Topic"] = "politics"
data3.columns = ["Doc Text", "Topic"]
data3.head()
```

Out[4]:

	Doc Text	Topic
0	Labour plans maternity pay rise\n\nMaternity p...	politics
1	Watchdog probes e-mail deletions\n\nThe inform...	politics
2	Hewitt decries 'career sexism'\n\nPlans to ext...	politics
3	Labour chooses Manchester\n\nThe Labour Party ...	politics
4	Brown ally rejects Budget spree\n\nChancellor ...	politics

In [5]:
```
file_list4 = glob.glob(os.path.join(os.getcwd(), r"C:\Users\kalya\OneDrive - University of Illinois at Chicago\!UIC\!!Data Science\!Project\News Articles\bbc-fulltext\bbc\sport/", "*.txt"))

#all_files = os.listdir(r"C:\Users\kalya\OneDrive - University of Illinois at Chicago\!UIC\Data Science\Project\News Articles\bbc-fulltext\bbc\business/")
# imagine you're one directory above test dir
corpus4 = []
for file_path in file_list4:
    with open(file_path) as f_input:
        corpus4.append(f_input.read())

data4 = pd.DataFrame(data = corpus4, columns = ["Doc Name"])
#data = data.reset_index()
data4["Topic"] = "sport"
data4.columns = ["Doc Text", "Topic"]
data4.head()
```

Out[5]:

	Doc Text	Topic
0	Claxton hunting first major medal\n\nBritish h...	sport
1	O'Sullivan could run in Worlds\n\nSonia O'Sull...	sport
2	Greene sets sights on world title\n\nMaurice G...	sport
3	IAAF launches fight against drugs\n\nThe IAAF ...	sport
4	Dibaba breaks 5,000m world record\n\nEthiopia'...	sport

In [6]:
```
file_list5 = glob.glob(os.path.join(os.getcwd(), r"C:\Users\kalya\OneDrive - U
niversity of Illinois at Chicago\!UIC\!!Data Science\!Project\News Articles\bb
c-fulltext\bbc\tech/", "*.txt"))

#all_files = os.listdir(r"C:\Users\kalya\OneDrive - University of Illinois at
 Chicago\!UIC\Data Science\Project\News Articles\bbc-fulltext\bbc\business/")
# imagine you're one directory above test dir
corpus5 = []
for file_path in file_list5:
    with open(file_path) as f_input:
        corpus5.append(f_input.read())

data5 = pd.DataFrame(data = corpus5, columns = ["Doc Name"])
#data = data.reset_index()
data5["Topic"] = "sport"
data5.columns = ["Doc Text", "Topic"]
data5.head()
```

Out[6]:

	Doc Text	Topic
0	Ink helps drive democracy in Asia\n\nThe Kyrgy...	sport
1	China net cafe culture crackdown\n\nChinese au...	sport
2	Microsoft seeking spyware trojan\n\nMicrosoft ...	sport
3	Digital guru floats sub-$100 PC\n\nNicholas Ne...	sport
4	Technology gets the creative bug\n\nThe hi-tec...	sport

In [7]:
```
d = [data1, data2, data3, data4,data5]
data = pd.concat(d)
data.to_csv("News.csv")
data.head()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2225 entries, 0 to 400
Data columns (total 2 columns):
Doc Text    2225 non-null object
Topic       2225 non-null object
dtypes: object(2)
memory usage: 52.1+ KB
```

In [8]:
```
data["Doc Text"] = data["Doc Text"].str.replace('\n\n', ' ')
data["Doc Text"] = data["Doc Text"].str.replace('[^\w\s]','')
data["Doc Text"] = data["Doc Text"].str.replace('\s+[a-zA-Z]\s+','')
data["Doc Text"] = data["Doc Text"].str.replace('\^[a-zA-Z]\s+','')

data["Doc Text"] = data["Doc Text"].str.replace('\s+',' ')
data.head()
```

Out[8]:

	Doc Text	Topic
0	Ad sales boost Time Warner profit Quarterly pr...	business
1	Dollar gains on Greenspan speech The dollar ha...	business
2	Yukos unit buyer faces loan claim The owners o...	business
3	High fuel prices hit BAs profits British Airwa...	business
4	Pernod takeover talk lifts Domecq Shares in UK...	business

In [9]:
```
data["Doc Text"] = data["Doc Text"].str.lower()
data.head()
```

Out[9]:

	Doc Text	Topic
0	ad sales boost time warner profit quarterly pr...	business
1	dollar gains on greenspan speech the dollar ha...	business
2	yukos unit buyer faces loan claim the owners o...	business
3	high fuel prices hit bas profits british airwa...	business
4	pernod takeover talk lifts domecq shares in uk...	business

In [10]:
```
# Install stopwords package
import nltk

nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\kalya\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [11]:
```
# Print list of stop words
stop_list = stopwords.words('english')
#print(stop_list)

# Remove stop words from text
data["Doc Text"] = data["Doc Text"].apply(lambda x: " ".join([word for word in
x.split() if word not in stop_list]))
data.head()
```

Out[11]:

	Doc Text	Topic
0	ad sales boost time warner profit quarterly pr...	business
1	dollar gains greenspan speech dollar hit highe...	business
2	yukos unit buyer faces loan claim owners embat...	business
3	high fuel prices hit bas profits british airwa...	business
4	pernod takeover talk lifts domecq shares uk dr...	business

In [12]:
```
st = PorterStemmer()
data["Doc Text"] = data["Doc Text"].apply(lambda x: " ".join([st.stem(y) for y
in x.split()]))
#apply(lambda x: [stemmer.stem(y) for y in x])
data.head()
```

Out[12]:

	Doc Text	Topic
0	ad sale boost time warner profit quarterli pro...	business
1	dollar gain greenspan speech dollar hit highes...	business
2	yuko unit buyer face loan claim owner embattl ...	business
3	high fuel price hit ba profit british airway b...	business
4	pernod takeov talk lift domecq share uk drink ...	business

In [13]:
```
#data['tokens'] = data["Doc Text"].apply(lambda x: x.split())
#data.head()
```

In [14]:
```
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

In [15]:
```
X = data.iloc[:, :1]
Y = data['Topic']
```

In [16]:
```
#y = pd.DataFrame(data = LabelEncoder().fit_transform(Y), columns= ["Topic"])
y = pd.DataFrame(data = Y, columns= ["Topic"])
```

In [17]:
```
X.head()
```

Out[17]:

	Doc Text
0	ad sale boost time warner profit quarterli pro...
1	dollar gain greenspan speech dollar hit highes...
2	yuko unit buyer face loan claim owner embattl ...
3	high fuel price hit ba profit british airway b...
4	pernod takeov talk lift domecq share uk drink ...

In [18]:
```
y.head()
```

Out[18]:

	Topic
0	business
1	business
2	business
3	business
4	business

In [19]:
```
y["Topic"].value_counts()
```

Out[19]:
```
sport            912
business         510
politics         417
entertainment    386
Name: Topic, dtype: int64
```

In [20]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [21]:
```
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(data["Doc Text"])
x1 = tfidf_matrix.toarray()
```

In [22]:
```
from sklearn.model_selection import train_test_split
x_train1, x_test, y_train1, y_test = train_test_split(x1, y, test_size = 0.25,
random_state = 100)
x_train, x_val, y_train, y_val = train_test_split(x_train1, y_train1, test_siz
e = 0.25, random_state = 100)
```

In [23]:
```
f =pd.DataFrame(data = x1)
f.head()

p =pd.DataFrame(data = x_train)
p.head()

p1 =pd.DataFrame(data = x_train1)
p1.head()

t = pd.DataFrame(data = x_test)
```

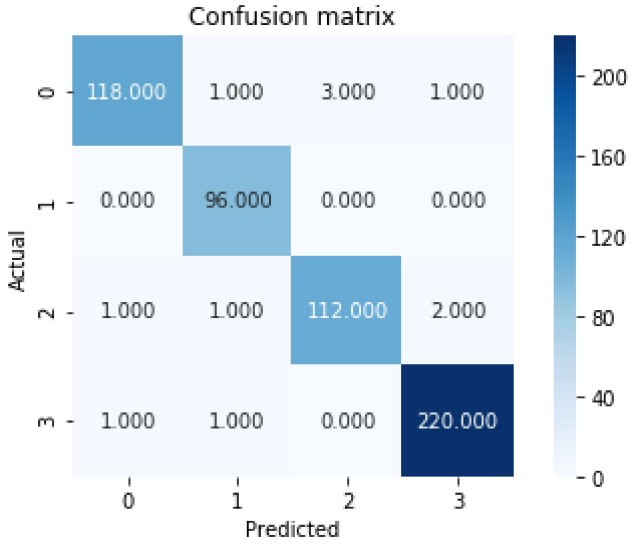
In [24]:
```
x_train.shape
```

Out[24]:
```
(1251, 39000)
```

In [25]:
```
# SVM
from sklearn.svm import LinearSVC
```

In [26]:
```
# SVM (Linear) with all variables

svm = LinearSVC()
print(cross_val_score(svm, x1,y.values.ravel(), cv = 5, scoring = 'f1_macro').
mean())

svm.fit(x_train1, y_train1.values.ravel())
y_pred = svm.predict(x_test)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

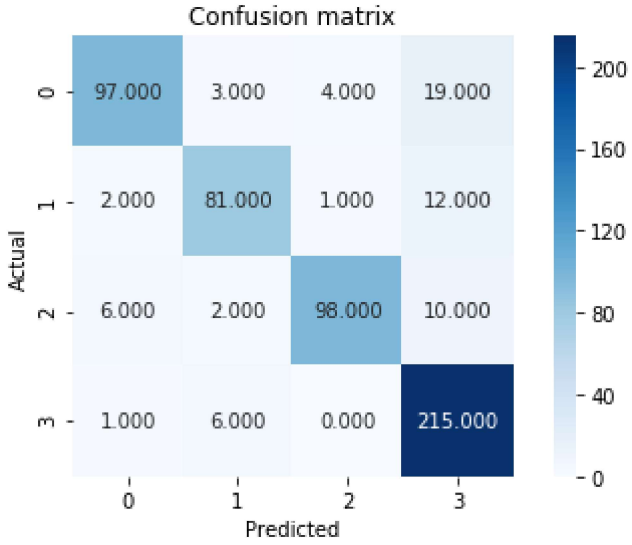
```
0.9642893919085964
[[118   1   3   1]
 [  0  96   0   0]
 [  1   1 112   2]
 [  1   1   0 220]]
0.9802513464991023
0.019748653500897717
[0.98333333 0.96969697 0.97391304 0.98654709]
[0.95934959 1.         0.96551724 0.99099099]
[0.97119342 0.98461538 0.96969697 0.98876404]
0.9785674537235087
```



In [27]:
```
# Naive Bayes

from sklearn.naive_bayes import GaussianNB
```

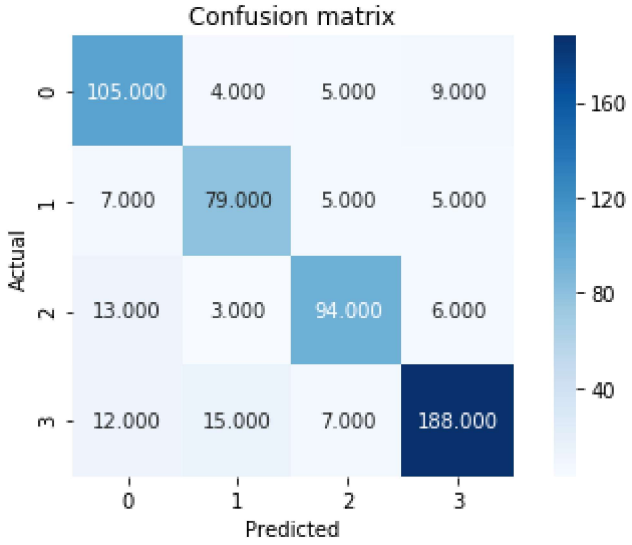
In [28]:
```
# Naive Bayes with all variables

gb = GaussianNB()
print(cross_val_score(gb, x1,y.values.ravel(), cv = 5, scoring = 'f1_macro').m
ean())

gb.fit(x_train1,y_train1.values.ravel())
y_pred = gb.predict(x_test)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.8446022412930663
[[ 97   3   4  19]
 [  2  81   1  12]
 [  6   2  98  10]
 [  1   6   0 215]]
0.881508078994614
0.11849192100538597
[0.91509434 0.88043478 0.95145631 0.83984375]
[0.78861789 0.84375    0.84482759 0.96846847]
[0.84716157 0.86170213 0.89497717 0.89958159]
0.8758556146549767
```



In [29]:
```
# Decision Tree with all variables

dt = DecisionTreeClassifier()
print(cross_val_score(dt, x1,y.values.ravel(), cv = 5, scoring = 'f1_macro').m
ean())

dt.fit(x_train1,y_train1.values.ravel())
y_pred = dt.predict(x_test)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.8046960138116523
[[105   4   5   9]
 [  7  79   5   5]
 [ 13   3  94   6]
 [ 12  15   7 188]]
0.8366247755834829
0.16337522441651708
[0.76642336 0.78217822 0.84684685 0.90384615]
[0.85365854 0.82291667 0.81034483 0.84684685]
[0.80769231 0.80203046 0.82819383 0.8744186 ]
0.8280838004488453
```



In [30]:
```
from sklearn.feature_selection import SelectKBest, f_classif
from math import sqrt
```

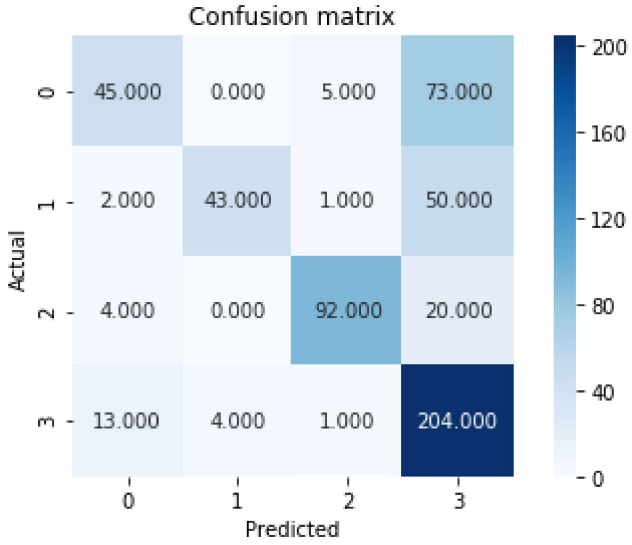
In [31]:
```
features = p.columns
features
bestfeatures = SelectKBest(score_func = f_classif, k=20)
#print(bestfeatures)

fit = bestfeatures.fit(x_train1, y_train1)
col = fit.get_support()
new_features = []
for bool, feature in zip(col, features):
    if bool:
        new_features.append(feature)

fit2 = bestfeatures.fit(f, y)
col2 = fit2.get_support()
new_features2 = []
for bool, feature in zip(col2, features):
    if bool:
        new_features2.append(feature)

#new_features
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724: D
ataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\feature_selection\univaria
te_selection.py:114: UserWarning: Features [   17    18    27 ... 38973 38976
38979] are constant.
  UserWarning)
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\feature_selection\univaria
te_selection.py:115: RuntimeWarning: invalid value encountered in true_divide
  f = msb / msw
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724: D
ataConversionWarning: A column-vector y was passed when a 1d array was expect
ed. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

In [32]:
```
orig= pd.DataFrame(f, columns=new_features2)
train = pd.DataFrame(p1, columns=new_features)


test = pd.DataFrame(data = t, columns = new_features)

#orig.head()
```

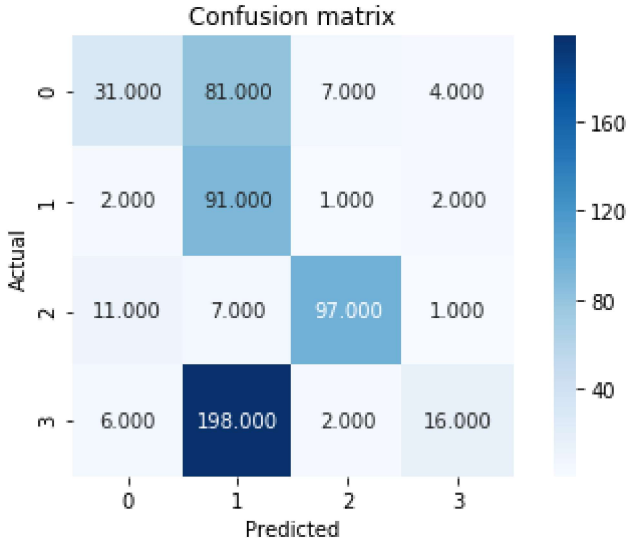
In [33]:
```
# SVM (Linear) with

svm = LinearSVC()
print(cross_val_score(svm, orig,y.values.ravel(), cv = 5, scoring = 'f1_macro'
).mean())

svm.fit(train, y_train1.values.ravel())
y_pred = svm.predict(test)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.6482234858091478
[[ 45   0   5  73]
 [  2  43   1  50]
 [  4   0  92  20]
 [ 13   4   1 204]]
0.6894075403949731
0.3105924596050269
[0.703125   0.91489362 0.92929293 0.58789625]
[0.36585366 0.44791667 0.79310345 0.91891892]
[0.48128342 0.6013986  0.85581395 0.71704745]
0.6638858572541156
```



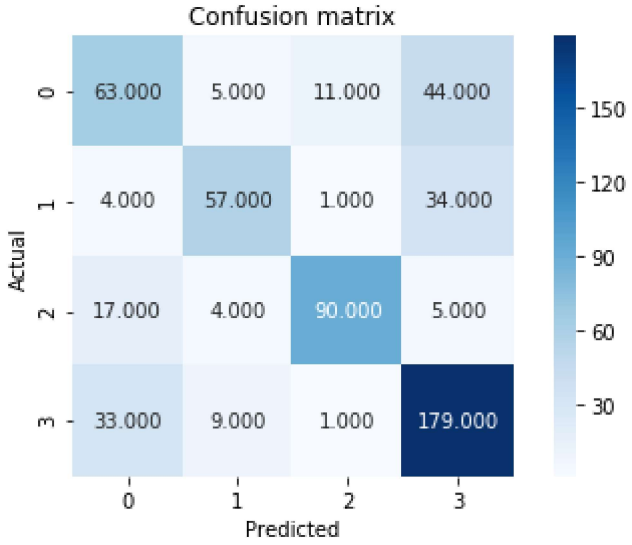
In [34]:
```
gb = GaussianNB()
print(cross_val_score(gb, orig,y.values.ravel(), cv = 5, scoring = 'f1_macro')
.mean())

gb.fit(train,y_train1.values.ravel())

#print(test.columns)
y_pred = gb.predict(test)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.6790885929965559
[[ 31  81   7   4]
 [  2  91   1   2]
 [ 11   7  97   1]
 [  6 198   2  16]]
0.42190305206463197
0.578096947935368
[0.62       0.24137931 0.90654206 0.69565217]
[0.25203252 0.94791667 0.8362069  0.07207207]
[0.3583815  0.38477801 0.86995516 0.13061224]
0.4359317293559487
```



In [35]:
```
# Decision Tree


dt = DecisionTreeClassifier()
print(cross_val_score(dt, orig,y.values.ravel(), cv = 5, scoring = 'f1_macro')
.mean())

dt.fit(train,y_train1.values.ravel())
test = pd.DataFrame(data = t, columns = new_features)
#print(test.columns)
y_pred = dt.predict(test)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.6676246593854498
[[ 63   5  11  44]
 [  4  57   1  34]
 [ 17   4  90   5]
 [ 33   9   1 179]]
0.6983842010771992
0.30161579892280077
[0.53846154 0.76       0.87378641 0.68320611]
[0.51219512 0.59375    0.77586207 0.80630631]
[0.525      0.66666667 0.82191781 0.73966942]
0.688313474093362
```



In [36]:
```
# features = p.columns
# features
# bestfeatures = SelectKBest(score_func = f_classif, k=None)
# #print(bestfeatures)

# fit = bestfeatures.fit(x_train1, y_train1)
# col = fit.get_support()
# new_features = []
# for bool, feature in zip(col, features):
#     if bool:
#         new_features.append(feature)

# #new_features
```

In [37]:
```
# The above feature selection took a lot of time
```

In [38]:
```
# from sklearn.feature_selection import RFECV

# model = ExtraTreesClassifier()
# rfecv  = RFECV(estimator = model, min_features_to_select = 2)
# rfecv.fit(x_train1,y_train1)

# col = rfecv.support_
# rank = rfecv.ranking_
# rank
# col


# new_features = []
# for bool, feature in zip(col, features):
#     if bool:
#         new_features.append(feature)
```

In [39]:
```
# The above feature selection took a lot of time
```

In [40]:
```
# train2 = pd.DataFrame(p1, columns=new_features)
# train2.head()
```

In [41]:
```
# With Bayesian Optimization and RF

# def objective_func(args):
#     n_estimators = args['n_estimators']
#     max_depth = args['max_depth']
#     min_samples_split = args['min_samples_split']
# #    criterion = args['criterion']
# #    max_features = args['max_features']
#     min_samples_leaf = args['min_samples_leaf']

#     #    clf = RandomForestClassifier(n_estimators = n_estimators, max_depth
= max_depth, criterion=criterion,
#  #                       max_features=max_features,min_samples_leaf=min_sampl
es_leaf, min_samples_split=min_samples_split)

#     clf = RandomForestClassifier(n_estimators = n_estimators, max_depth = ma
x_depth, min_samples_leaf=min_samples_leaf, min_samples_split=min_samples_spli
t, n_jobs = -1)

#     clf.fit(x_train,y_train.values.ravel())
#     y_pred = clf.predict(x_val)
#     f1 = -(f1_score(y_pred,y_val,  average='macro'))
#     return f1
# space = {'n_estimators': hp.choice('n_estimators', range(200, 1001)),
#          'max_depth': hp.choice('max_depth', range(20,101)),
# #         'criterion': hp.choice('criterion',['gini', 'entropy']),
# #         'max_features': hp.choice('max_features', ['auto', 'sqrt']),
#         'min_samples_split': hp.choice('min_samples_split', range(2,11)),
#          'min_samples_leaf' : hp.choice('min_samples_leaf',range(1,5))}



# best_classifier = fmin(objective_func, space, algo=tpe.suggest, max_evals=10
0)
# print(best_classifier)
```

In [42]:
```
# bc = RandomForestClassifier(max_depth = 26,min_samples_leaf = 2, min_samples
_split = 3, n_estimators = 175)
# bc.fit(x_train,y_train)
# y_pred = bc.predict(x_test)

# # Compute confusion matrix
# conf_matrix = metrics.confusion_matrix(y_test, y_pred)
# print(conf_matrix)

# # Plot confusion matrix
# sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = pl
t.cm.Blues)
# plt.ylabel('Actual')
# plt.xlabel('Predicted')
# plt.title('Confusion matrix')
# plt.tight_layout()

# # Compute evaluation metrics
# print(metrics.accuracy_score(y_test, y_pred)) # accuracy
# print(1 - metrics.accuracy_score(y_test, y_pred)) # error
# print(metrics.precision_score(y_test, y_pred, average = None)) # precision
# print(metrics.recall_score(y_test, y_pred, average = None)) # recall
# print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
# print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score averag
e
```

In [43]:
```
# THe above model took a lot of time to run
```

In [44]:
```
# WITH LDA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from math import sqrt
lda = LinearDiscriminantAnalysis(n_components = None)
x_lda = lda.fit_transform(x_train, y_train.values.ravel())
x_lda1 = lda.fit_transform(x_train1, y_train1.values.ravel())
x_vallda = lda.transform(x_val)
x_testlda = lda.transform(x_test)
x_lda = pd.DataFrame(data = x_lda)
```

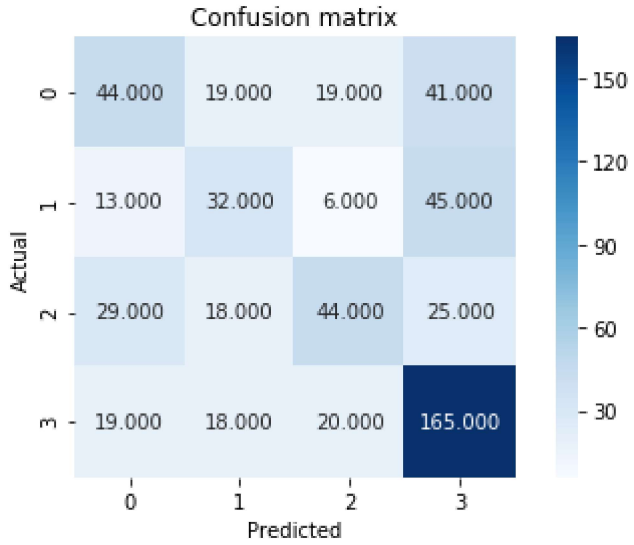
```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\discriminant_analysis.py:3
88: UserWarning: Variables are collinear.
  warnings.warn("Variables are collinear.")
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\discriminant_analysis.py:3
88: UserWarning: Variables are collinear.
  warnings.warn("Variables are collinear.")
```

In [45]:
```
# WIth LDA and Random Forest in Bayesian Optimization

def objective_func(args):

    n_estimators = args['n_estimators']
    max_depth = args['max_depth']
    min_samples_split = args['min_samples_split']
#    criterion = args['criterion']
#    max_features = args['max_features']
    min_samples_leaf = args['min_samples_leaf']

    #    clf = RandomForestClassifier(n_estimators = n_estimators, max_depth = max_depth, criterion=criterion,
 #                        max_features=max_features,min_samples_leaf=min_samples_leaf, min_samples_split=min_samples_split)

    clf = RandomForestClassifier(n_estimators = n_estimators, max_depth = max_depth, min_samples_leaf=min_samples_leaf, min_samples_split=min_samples_split, n_jobs = -1)

    #clf.fit(x_lda,y_train.values.ravel())
    #y_pred = clf.predict(x_vallda)
    error = 1-(cross_val_score(clf, x_lda1,y_train1.values.ravel(), cv = 5, scoring = 'f1_macro'))
    f1 = mean(error) + stdev(error)
    #print(f1)
    return f1
space = {'n_estimators': hp.choice('n_estimators', np.arange(200, 1001, step = 1)),
         'max_depth': hp.choice('max_depth', np.arange(20,101, step =1)),
#         'criterion': hp.choice('criterion',['gini', 'entropy']),
#         'max_features': hp.choice('max_features', ['auto', 'sqrt']),
        'min_samples_split': hp.choice('min_samples_split', np.arange(2,11, step =1)),
         'min_samples_leaf' : hp.choice('min_samples_leaf',np.arange(2,5, step =1))}



best_classifier = fmin(objective_func, space, algo=tpe.suggest, max_evals=100)
print(best_classifier)
```

```
100%|██████████| 100/100 [04:52<00:00,  2.57s/it, best loss: 0.0019438202212934488]
{'max_depth': 20, 'min_samples_leaf': 0, 'min_samples_split': 4, 'n_estimators': 677}
```

In [46]:
```
c = lambda b : 1 if b == 0 else b
d = lambda b : 2 if b <= 1 else b
bc = RandomForestClassifier(max_depth = c(best_classifier['max_depth']), min_s
amples_leaf = c(best_classifier['min_samples_leaf']), min_samples_split = d(be
st_classifier['min_samples_split'])
                           ,n_estimators = best_classifier['n_estimators'])

bc.fit(x_lda1, y_train1)

y_pred = bc.predict(x_testlda)
# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)


# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\ipykernel_launcher.py:6: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Plea
se change the shape of y to (n_samples,), for example using ravel().


[[ 44  19  19  41]
 [ 13  32   6  45]
 [ 29  18  44  25]
 [ 19  18  20 165]]
0.5116696588868941
0.4883303411131059
[0.41904762 0.36781609 0.49438202 0.59782609]
[0.35772358 0.33333333 0.37931034 0.74324324]
[0.38596491 0.34972678 0.42926829 0.6626506 ]
0.4569026458323878
```

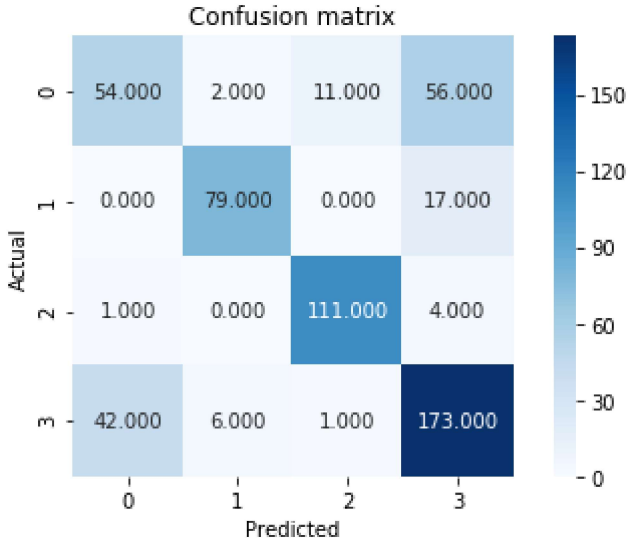


In [47]:
```
# LDA didn't perform well as the variables are collinear
```

In [48]:
```
# With PCA
from sklearn.decomposition import PCA

# pca = PCA(n_components=3)
# x_pca = pca.fit_transform(x_train)
# x_valpca = pca.transform(x_val)
# x_testpca = pca.transform(x_test)
# print(pca.explained_variance_ratio_)

pca2 = PCA(n_components=3)
x_orig = pca2.fit_transform(f)
x_pca2 = pca2.fit_transform(x_train1)
x_testpca2 = pca2.transform(x_test)
#print(pca2.explained_variance_ratio_)

#x_testpca.shape
```

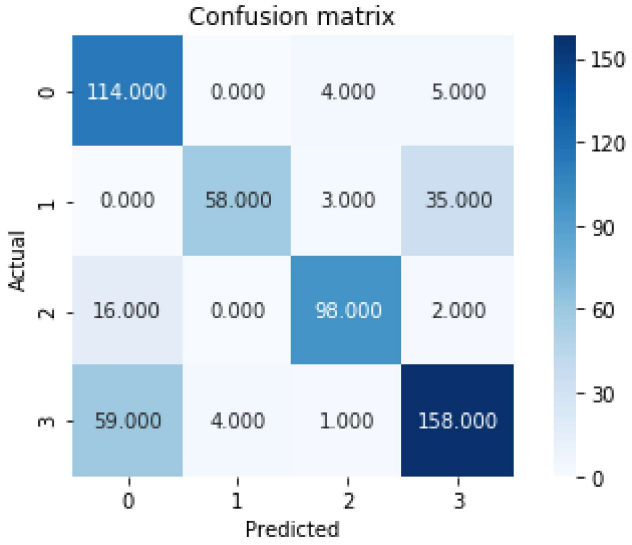
In [49]:
```
# SVM (Linear) with PCA

svm = LinearSVC()
print(cross_val_score(svm, x_orig,y.values.ravel(), cv = 5, scoring = 'f1_macr
o').mean())

svm.fit(x_pca2, y_train1.values.ravel())
y_pred = svm.predict(x_testpca2)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.7378424515784416
[[ 54   2  11  56]
 [  0  79   0  17]
 [  1   0 111   4]
 [ 42   6   1 173]]
0.748653500897666
0.251346499102334
[0.55670103 0.90804598 0.90243902 0.692     ]
[0.43902439 0.82291667 0.95689655 0.77927928]
[0.49090909 0.86338798 0.92887029 0.73305085]
0.7540545523489559
```



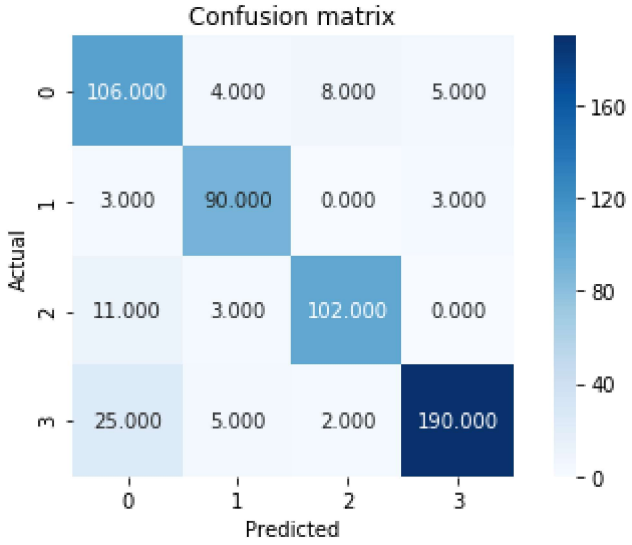
In [50]:
```
# Naive Bayes with principal components

gb = GaussianNB()
print(cross_val_score(gb, x_orig,y.values.ravel(), cv = 5, scoring = 'f1_macr
o').mean())

gb.fit(x_pca2,y_train1.values.ravel())
y_pred = gb.predict(x_testpca2)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.8021381042945016
[[114   0   4   5]
 [  0  58   3  35]
 [ 16   0  98   2]
 [ 59   4   1 158]]
0.7684021543985637
0.23159784560143626
[0.6031746  0.93548387 0.9245283  0.79      ]
[0.92682927 0.60416667 0.84482759 0.71171171]
[0.73076923 0.73417722 0.88288288 0.74881517]
0.774161123679691
```



In [51]:
```
# Decision Tree with principal components

dt = DecisionTreeClassifier()
print(cross_val_score(dt, x_orig,y.values.ravel(), cv = 5, scoring = 'f1_macr
o').mean())

dt.fit(x_pca2,y_train1.values.ravel())

y_pred = dt.predict(x_testpca2)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.893320642049467
[[106   4   8   5]
 [  3  90   0   3]
 [ 11   3 102   0]
 [ 25   5   2 190]]
0.8761220825852782
0.12387791741472176
[0.73103448 0.88235294 0.91071429 0.95959596]
[0.86178862 0.9375     0.87931034 0.85585586]
[0.79104478 0.90909091 0.89473684 0.9047619 ]
0.87490860801937
```

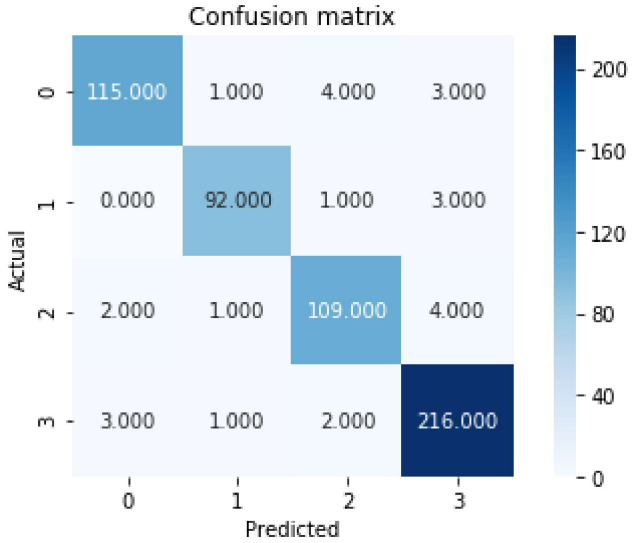


In [52]:
```
# With PCA, Bayesian Optimization, and SVM

def objective_func(args):
    C = args['C']
    pca = PCA(n_components=args['n'])
    x_pca1 = pca.fit_transform(x_train1)


    clf = LinearSVC(C= C)

    #clf.fit(x_pca,y_train.values.ravel())
    #y_pred = clf.predict(x_valpca)
    #f1 = -(f1_score(y_pred,y_val,  average='macro'))
    error = 1-(cross_val_score(clf, x_pca1,y_train1.values.ravel(), cv = 5, sc
oring = 'f1_macro'))
    f1 = mean(error) + stdev(error)
    #print(f1)
    return f1
space = {'C': hp.choice('C', np.arange(1,10, step = 1)),
         'n': hp.choice('n', np.arange(3,20, step =1))}



best_classifier = fmin(objective_func, space, algo=tpe.suggest, max_evals=100)
print(best_classifier)
```

```
100%|██████████| 100/100 [05:24<00:00,  3.59s/it, best loss: 0.04269031232647
185]
{'C': 5, 'n': 16}
```

In [53]:
```
pca = PCA(n_components=best_classifier['n'])
x_pca1 = pca.fit_transform(x_train1)
x_testpca = pca.transform(x_test)


bh = LinearSVC(C = best_classifier['C'])
bh.fit(x_pca1,y_train1.values.ravel())
y_pred = bh.predict(x_testpca)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
[[115   1   4   3]
 [  0  92   1   3]
 [  2   1 109   4]
 [  3   1   2 216]]
0.9551166965888689
0.044883303411131115
[0.95833333 0.96842105 0.93965517 0.95575221]
[0.93495935 0.95833333 0.93965517 0.97297297]
[0.94650206 0.96335079 0.93965517 0.96428571]
0.9534484324132476
```

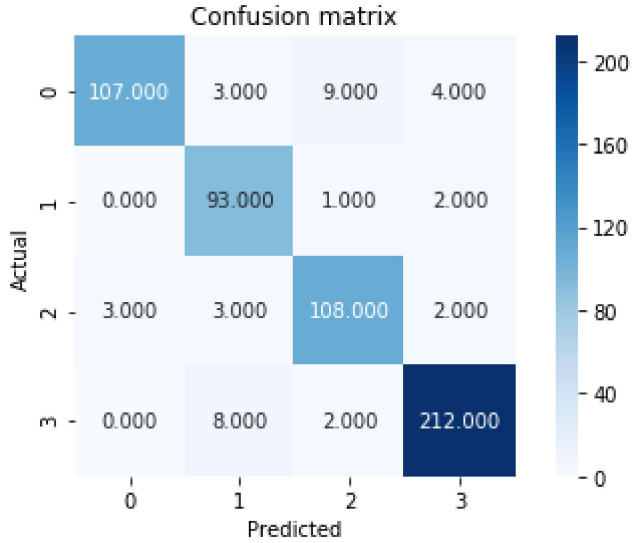


In [54]:
```
# With PCA, Bayesian Optimization, and Decision Tree

def objective_func(args):
    max_depth = args['max_depth']
    min_samples_split = args['min_samples_split']
    min_samples_leaf = args['min_samples_leaf']
    pca = PCA(n_components=args['n'])
    x_pca1 = pca.fit_transform(x_train1)


    clf = DecisionTreeClassifier(max_depth = max_depth, min_samples_leaf=min_s
amples_leaf, min_samples_split=min_samples_split)

    #clf.fit(x_pca,y_train.values.ravel())
    #y_pred = clf.predict(x_valpca)
    #f1 = -(f1_score(y_pred,y_val,  average='macro'))
    error = 1-(cross_val_score(clf, x_pca1,y_train1.values.ravel(), cv = 5, sc
oring = 'f1_macro'))
    f1 = mean(error) + stdev(error)
    #print(f1)
    return f1
space = {
         'max_depth': hp.choice('max_depth', np.arange(20,101, step =1)),
#         'criterion': hp.choice('criterion',['gini', 'entropy']),
#         'max_features': hp.choice('max_features', ['auto', 'sqrt']),
        'min_samples_split': hp.choice('min_samples_split', np.arange(2,12, st
ep =1)),
         'min_samples_leaf' : hp.choice('min_samples_leaf',np.arange(1,12, ste
p =1)),
        'n': hp.choice('n', np.arange(3,20, step =1))}



best_classifier = fmin(objective_func, space, algo=tpe.suggest, max_evals=100)
print(best_classifier)
```

```
100%|██████████| 100/100 [07:27<00:00,  3.70s/it, best loss: 0.07919763523353
167]
{'max_depth': 12, 'min_samples_leaf': 6, 'min_samples_split': 0, 'n': 5}
```

In [55]:
```
pca = PCA(n_components=best_classifier['n'])
x_pca1 = pca.fit_transform(x_train1)
x_testpca = pca.transform(x_test)

c = lambda a: 1 if a == 0 else a
d = lambda a: 2 if a<= 1 else a
bh = DecisionTreeClassifier(max_depth = c(best_classifier['max_depth']), min_s
amples_leaf=c(best_classifier['min_samples_leaf']), min_samples_split=d(best_c
lassifier['min_samples_split']))
bh.fit(x_pca1,y_train1.values.ravel())
y_pred = bh.predict(x_testpca)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
[[107   3   9   4]
 [  0  93   1   2]
 [  3   3 108   2]
 [  0   8   2 212]]
0.933572710951526
0.06642728904847395
[0.97272727 0.86915888 0.9        0.96363636]
[0.8699187  0.96875    0.93103448 0.95495495]
[0.91845494 0.91625616 0.91525424 0.95927602]
0.9273103371613671
```



In [56]:
```
# KNN
from sklearn.neighbors import KNeighborsClassifier
```

In [57]:
```
# WIth PCA , Bayesian Optimization and KNN
def objective_func(args):
    n_neighbors = args['n_neighbors']
    metric = args['metric']
    pca = PCA(n_components=args['n'])
    x_pca1 = pca.fit_transform(x_train1)

    clf = KNeighborsClassifier(n_neighbors=n_neighbors, metric=metric, n_jobs
= -1)

    #clf.fit(x_pca,y_train.values.ravel())
    #y_pred = clf.predict(x_valpca)
    #f1 = -(f1_score(y_pred,y_val,  average='macro'))
    error = 1-(cross_val_score(clf, x_pca1,y_train1.values.ravel(), cv = 5, sc
oring = 'f1_macro'))
    f1 = mean(error) + stdev(error)
    #print(f1)
    return f1
space = {'n_neighbors': hp.choice('n_neighbors',np.arange(1,1000, step =1)),
        'metric':hp.choice('metric', ["euclidean","manhattan"]),
        'n': hp.choice('n', np.arange(3,20, step =1))}

best_classifier = fmin(objective_func, space, algo=tpe.suggest, max_evals=100)
print(best_classifier)
```

```
  1%|          | 1/100 [00:07<11:48,  7.16s/it, best loss: 0.0679197062520172
6]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


  2%|▏         | 2/100 [00:11<10:28,  6.42s/it, best loss: 0.0679197062520172
6]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


  4%|▍         | 4/100 [00:19<08:16,  5.17s/it, best loss: 0.0679197062520172
6]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


  5%|▌         | 5/100 [00:24<07:55,  5.01s/it, best loss: 0.0679197062520172
6]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


  8%|▊         | 8/100 [00:36<06:35,  4.30s/it, best loss: 0.0679197062520172
6]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


  9%|█         | 9/100 [00:40<06:06,  4.03s/it, best loss: 0.0679197062520172
6]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 10%|█         | 10/100 [00:45<06:34,  4.38s/it, best loss: 0.067919706252017
26]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 15%|█         | 15/100 [01:05<05:51,  4.14s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 25%|██        | 25/100 [01:43<04:34,  3.66s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 27%|██▋       | 27/100 [01:51<04:38,  3.81s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 30%|███       | 30/100 [02:04<04:51,  4.17s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 31%|███       | 31/100 [02:08<04:52,  4.24s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 37%|███▋      | 37/100 [02:32<04:03,  3.86s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 38%|███▊      | 38/100 [02:37<04:24,  4.27s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 41%|████      | 41/100 [02:49<03:58,  4.05s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 43%|████▎     | 43/100 [02:58<04:01,  4.24s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 44%|████▍     | 44/100 [03:03<04:07,  4.42s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 46%|████▋     | 46/100 [03:11<03:50,  4.26s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 48%|████▊     | 48/100 [03:20<03:39,  4.22s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 49%|████▉     | 49/100 [03:25<03:43,  4.39s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 50%|█████     | 50/100 [03:29<03:43,  4.47s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 53%|█████▎    | 53/100 [03:41<03:12,  4.09s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 54%|█████▍    | 54/100 [03:46<03:10,  4.13s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 56%|█████▌    | 56/100 [03:54<02:57,  4.03s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 59%|█████▉    | 59/100 [04:06<02:39,  3.90s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 62%|██████    | 62/100 [04:19<02:42,  4.28s/it, best loss: 0.054937275757351
25]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 63%|██████▎   | 63/100 [04:23<02:33,  4.16s/it, best loss: 0.054937275757351
25]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 68%|██████▊   | 68/100 [04:44<02:07,  3.99s/it, best loss: 0.045034334794852
235]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 69%|██████▉   | 69/100 [04:49<02:12,  4.26s/it, best loss: 0.045034334794852
235]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 70%|███████   | 70/100 [04:53<02:11,  4.39s/it, best loss: 0.045034334794852
235]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 72%|███████▏  | 72/100 [05:02<01:56,  4.16s/it, best loss: 0.045034334794852
235]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 77%|███████▋  | 77/100 [05:19<01:21,  3.56s/it, best loss: 0.045034334794852
235]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 78%|███████▊  | 78/100 [05:24<01:27,  3.98s/it, best loss: 0.045034334794852
235]
```

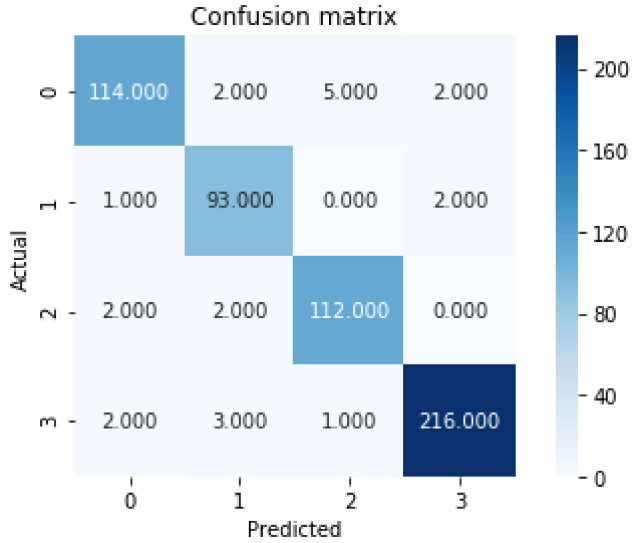
```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

 83%|████████▎ | 83/100 [05:44<01:05,  3.87s/it, best loss: 0.045034334794852
235]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

 85%|████████▌ | 85/100 [05:52<01:02,  4.13s/it, best loss: 0.045034334794852
235]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 86%|████████▌ | 86/100 [05:57<01:00,  4.33s/it, best loss: 0.045034334794852
235]

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


 92%|█████████▏| 92/100 [06:20<00:29,  3.73s/it, best loss: 0.045034334794852
235]
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

C:\Users\kalya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:
1437: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in
labels with no predicted samples.
  'precision', 'predicted', average, warn_for)


100%|██████████| 100/100 [06:51<00:00,  3.81s/it, best loss: 0.04503433479485
2235]
{'metric': 0, 'n': 6, 'n_neighbors': 24}
```

In [58]:
```
pca = PCA(n_components=best_classifier['n'])
x_pca1 = pca.fit_transform(x_train1)
#x_valpca = pca.transform(x_val)
x_testpca = pca.transform(x_test)

c = lambda a: 'euclidean' if a == 0 else 'manhattan'
bh = KNeighborsClassifier(metric = c(best_classifier['metric']), n_neighbors=
best_classifier['n_neighbors'])
bh.fit(x_pca1,y_train1.values.ravel())
y_pred = bh.predict(x_testpca)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

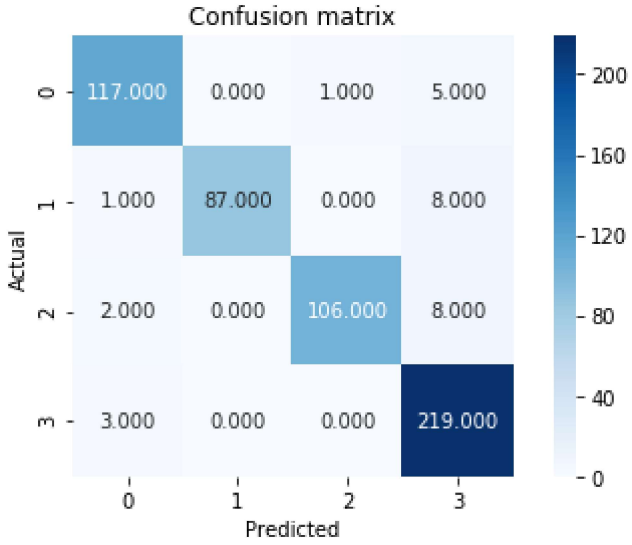
```
[[114   2   5   2]
 [  1  93   0   2]
 [  2   2 112   0]
 [  2   3   1 216]]
0.9605026929982047
0.03949730700179532
[0.95798319 0.93       0.94915254 0.98181818]
[0.92682927 0.96875    0.96551724 0.97297297]
[0.94214876 0.94897959 0.95726496 0.97737557]
0.9564422187607826
```



In [59]:
```
#Ensemble Methods
```

In [60]:
```
# Random Forest with all variables
bc = RandomForestClassifier(n_estimators = 500)
print(cross_val_score(bc, x1, y.values.ravel(), cv = 5, scoring = 'f1_macro').
mean())

bc.fit(x_train1,y_train1.values.ravel())
y_pred = bc.predict(x_test)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
0.9225939048500698
[[117   0   1   5]
 [  1  87   0   8]
 [  2   0 106   8]
 [  3   0   0 219]]
0.9497307001795332
0.050269299820466795
[0.95121951 1.         0.99065421 0.9125    ]
[0.95121951 0.90625    0.9137931  0.98648649]
[0.95121951 0.95081967 0.95067265 0.94805195]
0.9501909445295319
```

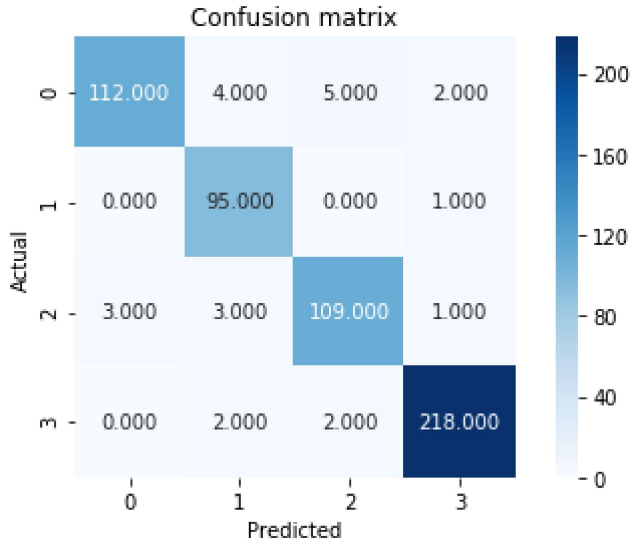


In [61]:
```
# With PCA, Bayesian Optimization, and Random Forest

def objective_func(args):
    n_estimators = args['n_estimators']
    max_depth = args['max_depth']
    min_samples_split = args['min_samples_split']
    min_samples_leaf = args['min_samples_leaf']
    pca = PCA(n_components=args['n'])
    x_pca1 = pca.fit_transform(x_train1)
    x_testpca = pca.transform(x_test)

    clf = RandomForestClassifier(n_estimators = n_estimators, max_depth = max_
depth, min_samples_leaf=min_samples_leaf, min_samples_split=min_samples_split,
n_jobs = -1)

    #clf.fit(x_pca,y_train.values.ravel())
    #y_pred = clf.predict(x_valpca)
    #f1 = -(f1_score(y_pred,y_val,  average='macro'))
    error = 1-(cross_val_score(clf, x_pca1,y_train1.values.ravel(), cv = 5, sc
oring = 'f1_macro'))
    f1 = mean(error) + stdev(error)
    #print(f1)
    return f1
space = {'n_estimators': hp.choice('n_estimators', np.arange(20,1001, step =1
)),
         'max_depth': hp.choice('max_depth', np.arange(20,101, step =1)),
#          'criterion': hp.choice('criterion',['gini', 'entropy']),
#          'max_features': hp.choice('max_features', ['auto', 'sqrt']),
        'min_samples_split': hp.choice('min_samples_split', np.arange(2,12, st
ep =1)),
         'min_samples_leaf' : hp.choice('min_samples_leaf',np.arange(1,12, ste
p =1)),
        'n': hp.choice('n', np.arange(3,20, step =1))}



best_classifier = fmin(objective_func, space, algo=tpe.suggest, max_evals=100)
print(best_classifier)
```

```
100%|██████████| 100/100 [14:48<00:00,  9.86s/it, best loss: 0.04040873200003
8965]
{'max_depth': 60, 'min_samples_leaf': 0, 'min_samples_split': 4, 'n': 12, 'n_
estimators': 101}
```

In [62]:
```
pca = PCA(n_components=best_classifier['n'])
x_pca1 = pca.fit_transform(x_train1)
#x_valpca = pca.transform(x_val)
x_testpca = pca.transform(x_test)


c = lambda b : 1 if b == 0 else b
d = lambda b : 2 if b <= 1 else b
bc = RandomForestClassifier(max_depth = c(best_classifier['max_depth']), min_s
amples_leaf = c(best_classifier['min_samples_leaf']), min_samples_split = d(be
st_classifier['min_samples_split'])
                            ,n_estimators = best_classifier['n_estimators'])

bc.fit(x_pca1, y_train1)

y_pred = bc.predict(x_testpca)
# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)


# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
C:\Users\kalya\Anaconda3\lib\site-packages\ipykernel_launcher.py:12: DataConv
ersionWarning: A column-vector y was passed when a 1d array was expected. Ple
ase change the shape of y to (n_samples,), for example using ravel().
  if sys.path[0] == '':

[[112   4   5   2]
 [  0  95   0   1]
 [  3   3 109   1]
 [  0   2   2 218]]
0.9587073608617595
0.04129263913824055
[0.97391304 0.91346154 0.93965517 0.98198198]
[0.91056911 0.98958333 0.93965517 0.98198198]
[0.94117647 0.95       0.93965517 0.98198198]
0.9532034062460026
```



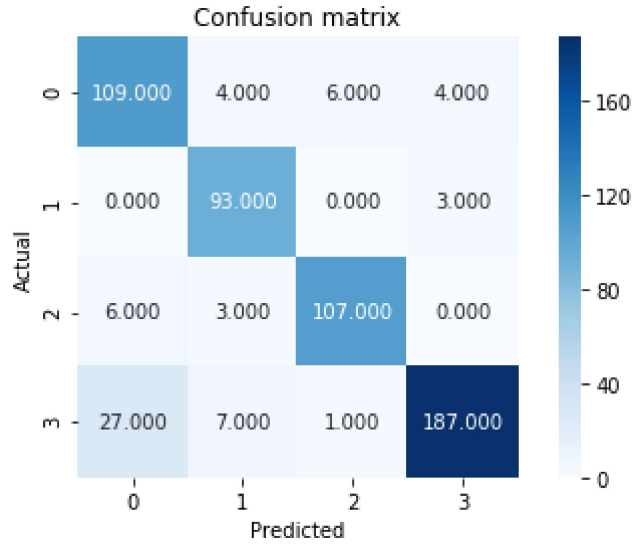
In [63]:
```
# Using Light GB

import lightgbm as lgbm
from lightgbm.sklearn import LGBMClassifier

from sklearn.model_selection import RandomizedSearchCV
```

In [64]:
```
# Using PCA, Random Search, and LGB

param = {
    'n_estimators':np.arange(200,1000, step =1),
    'learning_rate': np.linspace(0.1,0.7, num=7),
    'max_depth': np.arange(10,1000, step =1),
     'subsample': np.linspace(0.5,0.8, num=4),
    'min_split_gain': np.arange(0,6, step =1)}

bc1 = LGBMClassifier(objective = 'multiclass', random_state = 101)
rs = RandomizedSearchCV(estimator = bc1, param_distributions = param, n_iter=5
0,
                                    scoring='f1_macro',
                                    cv=3,
                                    verbose=1, random_state=8)

rs.fit(x_pca2, y_train1.values.ravel())

print(rs.best_params_)
print(rs.best_score_)
bcc = rs.best_estimator_
```

```
Fitting 3 folds for each of 50 candidates, totalling 150 fits

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent worke
rs.
[Parallel(n_jobs=1)]: Done 150 out of 150 | elapsed:   54.8s finished

{'subsample': 0.7000000000000001, 'n_estimators': 268, 'min_split_gain': 2,
'max_depth': 624, 'learning_rate': 0.6}
0.9033732638679315
```

In [65]:
```
bcc.fit(x_pca2,y_train1.values.ravel())
y_pred = bcc.predict(x_testpca2)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
[[109   4   6   4]
 [  0  93   0   3]
 [  6   3 107   0]
 [ 27   7   1 187]]
0.8904847396768402
0.10951526032315984
[0.76760563 0.86915888 0.93859649 0.96391753]
[0.88617886 0.96875    0.92241379 0.84234234]
[0.82264151 0.91625616 0.93043478 0.89903846]
0.8920927278041468
```

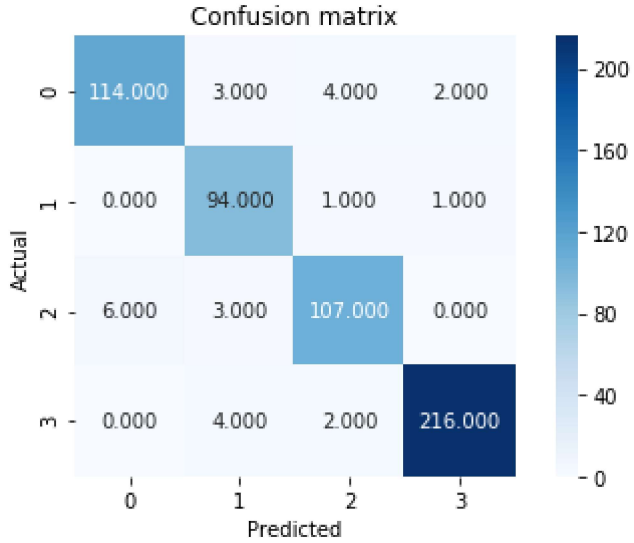


In [66]:
```
# # Using PCA, Bayesian Optimization, and LGB

def objective_func(args):
    n_estimators = args['n_estimators']
    max_depth = args['max_depth']
    min_split_gain = args['min_split_gain']
    learning_rate = args['learning_rate']
    subsample = args['subsample']

    pca = PCA(n_components=args['n'])
    x_pca1 = pca.fit_transform(x_train1)


    clf = LGBMClassifier(n_estimators = n_estimators, min_split_gain=min_split
_gain, subsample=subsample, max_depth= max_depth,learning_rate = learning_rate
)


#     clf.fit(x_pca1,y_train.values.ravel())
#     y_pred = clf.predict(x_valpca)
    error = 1-(cross_val_score(clf, x_pca1,y_train1.values.ravel(), cv = 5, sc
oring = 'f1_macro'))
    f1 = mean(error) + stdev(error)
    #print(f1)
    return f1
space = {'n_estimators': hp.choice('n_estimators',np.arange(200,1000, step =1
)),
        'min_split_gain': hp.choice('min_split_gain',np.arange(0,10, step =1
)),
         'subsample' : hp.uniform('subsample', 0.5, 0.9),
        'max_depth': hp.choice('max_depth',np.arange(10,1000, step =1)),
        'learning_rate': hp.uniform('learning_rate', 0.1, 0.5),
        'n': hp.choice('n', np.arange(3,20, step =1))}



best_classifier = fmin(objective_func, space, algo=tpe.suggest, max_evals=100)
print(best_classifier)
```

```
100%|██████████| 100/100 [10:57<00:00,  8.20s/it, best loss: 0.03820980781152
045]
{'learning_rate': 0.2428260562383549, 'max_depth': 340, 'min_split_gain': 0,
'n': 15, 'n_estimators': 316, 'subsample': 0.8272953669260521}
```

In [68]:
```
pca = PCA(n_components=best_classifier['n'])
x_pca1 = pca.fit_transform(x_train1)
x_testpca = pca.transform(x_test)

#c = lambda b : 1 if b == 0 else b
bccbo = LGBMClassifier(learning_rate= best_classifier['learning_rate'], n_esti
mators = best_classifier['n_estimators'], max_depth= c(best_classifier['max_de
pth']),
                       min_split_gain= best_classifier['min_split_gain'], subs
ample = best_classifier['subsample'])
bccbo.fit(x_pca1, y_train1.values.ravel())

y_pred = bccbo.predict(x_testpca)

# Compute confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(conf_matrix)

# Plot confusion matrix
sns.heatmap(conf_matrix, annot = True, fmt = ".3f", square = True, cmap = plt.
cm.Blues)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion matrix')
plt.tight_layout()

# Compute evaluation metrics
print(metrics.accuracy_score(y_test, y_pred)) # accuracy
print(1 - metrics.accuracy_score(y_test, y_pred)) # error
print(metrics.precision_score(y_test, y_pred, average = None)) # precision
print(metrics.recall_score(y_test, y_pred, average = None)) # recall
print(metrics.f1_score(y_test, y_pred, average = None)) # F1 score
print(metrics.f1_score(y_test, y_pred, average = 'macro')) # F1 score average
```

```
[[114   3   4   2]
 [  0  94   1   1]
 [  6   3 107   0]
 [  0   4   2 216]]
0.9533213644524237
0.04667863554757634
[0.95       0.90384615 0.93859649 0.98630137]
[0.92682927 0.97916667 0.92241379 0.97297297]
[0.9382716  0.94       0.93043478 0.97959184]
0.9470745560704154
```



In [ ]: