Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

«Прикладные интеллектуальные системы и экспертные системы»

Студент	Курдюков И.Ю.
Группа М-ИАП-23-1	
Руководитель	Кургасов В. В.
доцент, канд. пед. наук	

Липецк 2023 г.

Цель работы:

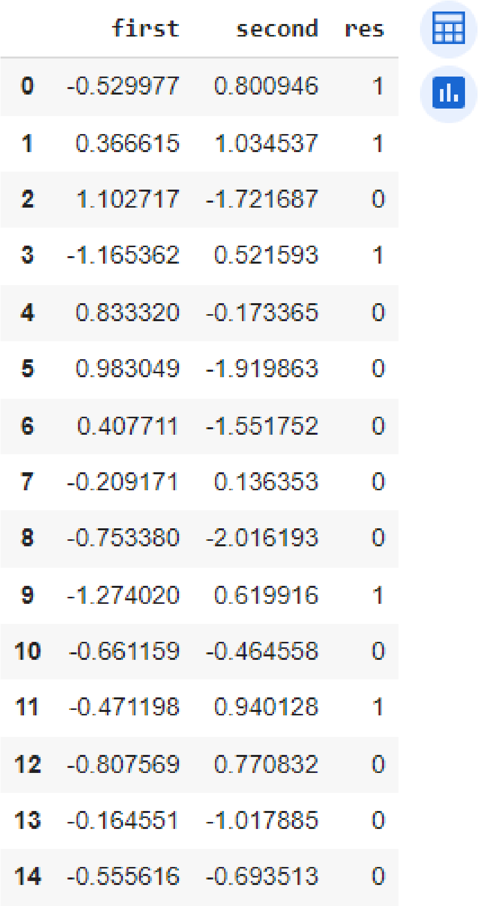
Получить практические навыки решения задачи бинарной классификации
данных в среде Jupiter Notebook. Научиться загружать данные, обучать классификаторы и проводить классификацию. Научиться оценивать точность полученных моделей.

Ход работы

Импортируем необходимые модули и библиотеки

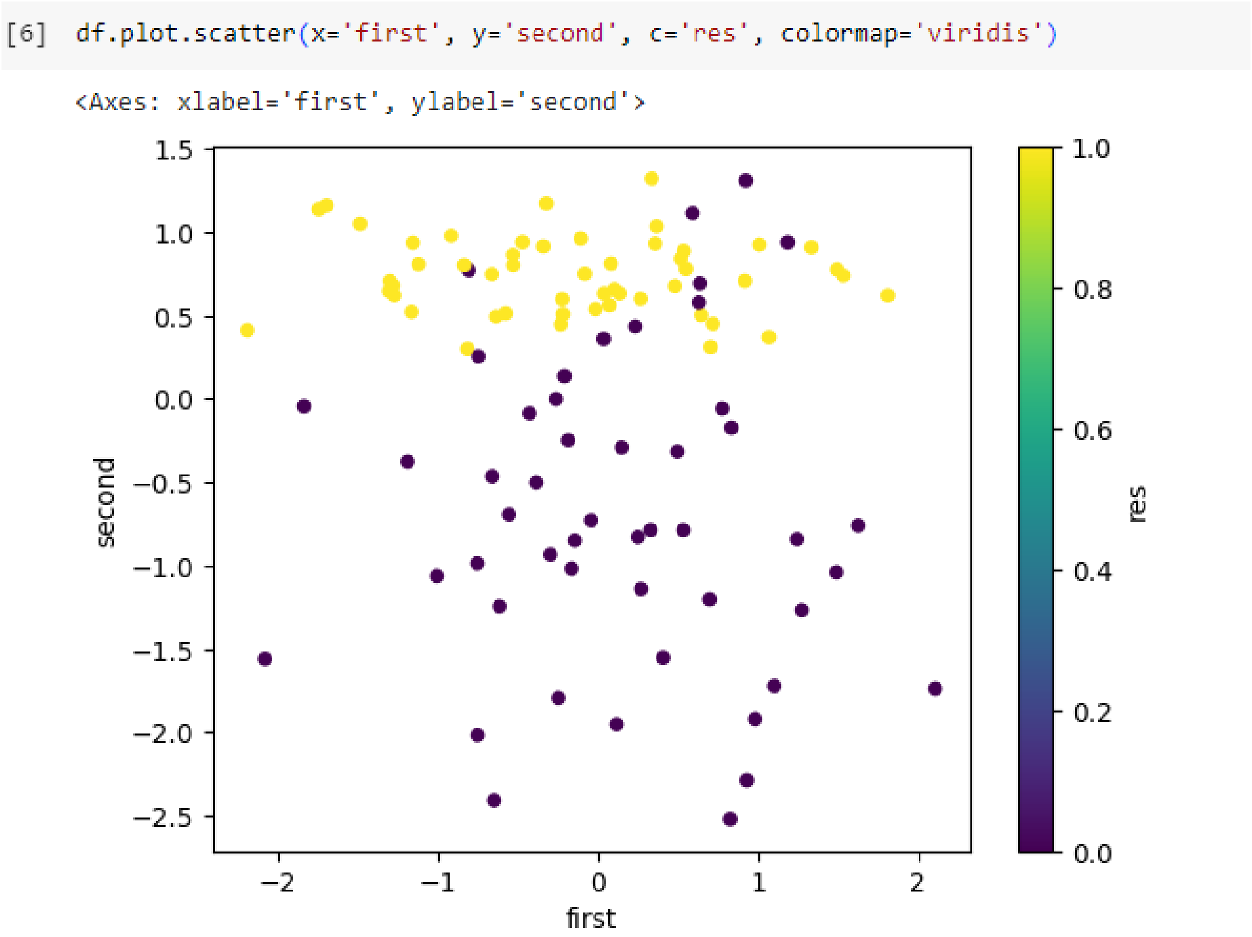
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
import numpy as np
```

```
ds = make_classification(random_state=58, n_informative=1, n_redundant=0,
n_features=2, n_clusters_per_class=1,
                         class_sep=0.7)
df = pd.DataFrame(ds[0], columns=['first', 'second'])
df['res'] = ds[1]
df.head(15)
```



	first	second	res
0	-0.529977	0.800946	1
1	0.366615	1.034537	1
2	1.102717	-1.721687	0
3	-1.165362	0.521593	1
4	0.833320	-0.173365	0
5	0.983049	-1.919863	0
6	0.407711	-1.551752	0
7	-0.209171	0.136353	0
8	-0.753380	-2.016193	0
9	-1.274020	0.619916	1
10	-0.661159	-0.464558	0
11	-0.471198	0.940128	1
12	-0.807569	0.770832	0
13	-0.164551	-1.017885	0
14	-0.555616	-0.693513	0

Построим график, отображающий нашу выборку

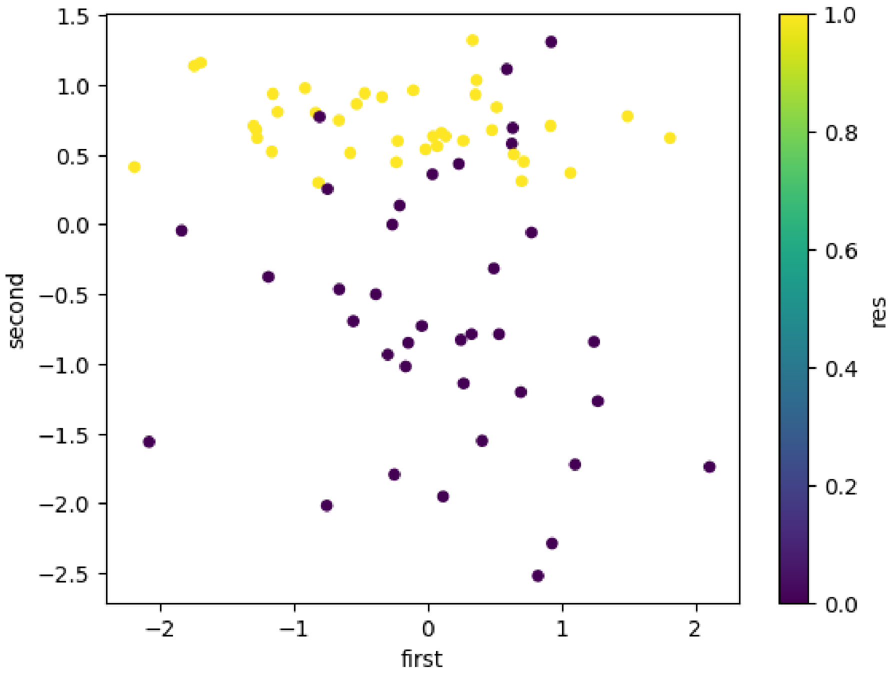


```
[6]  df.plot.scatter(x='first', y='second', c='res', colormap='viridis')
```

```
<Axes: xlabel='first', ylabel='second'>
```

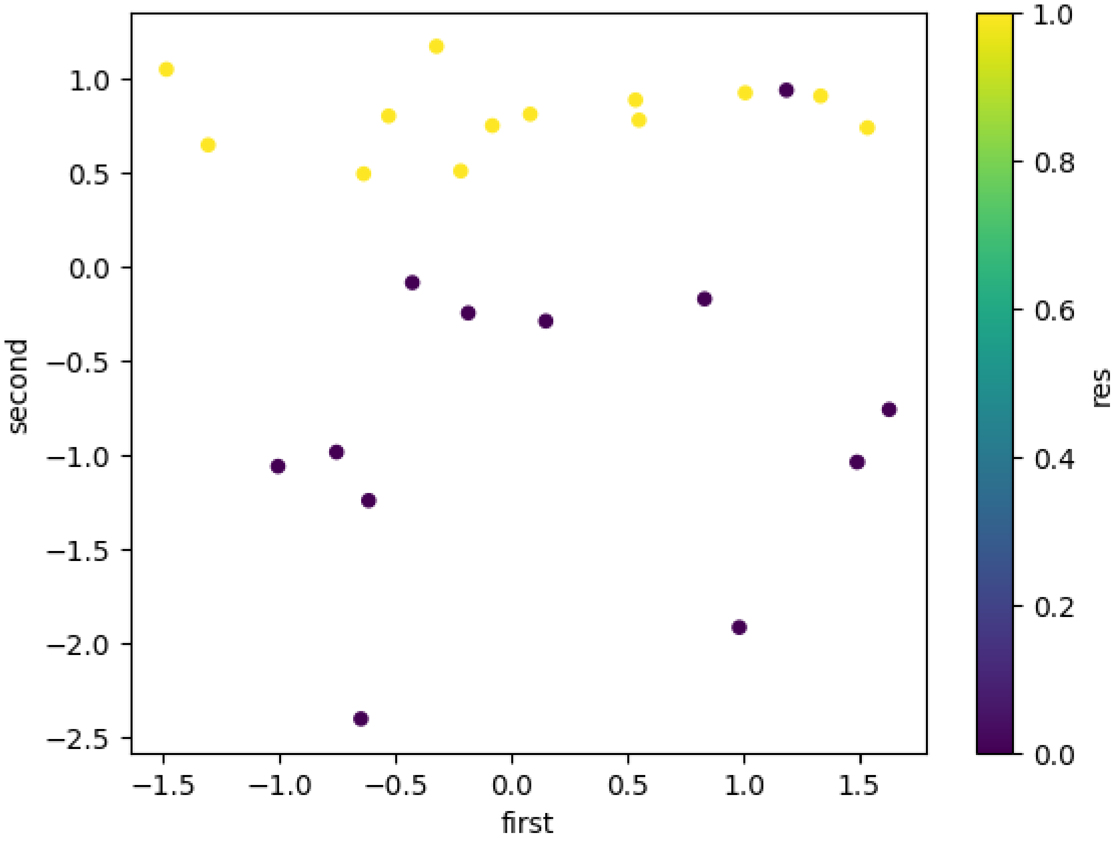
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(ds[0], ds[1])
train_df = pd.DataFrame(x_train, columns=['first', 'second'])
train_df['res'] = y_train
train_df.plot.scatter(x='first', y='second', c='res', colormap='viridis')
```

```
<Axes: xlabel='first', ylabel='second'>
```



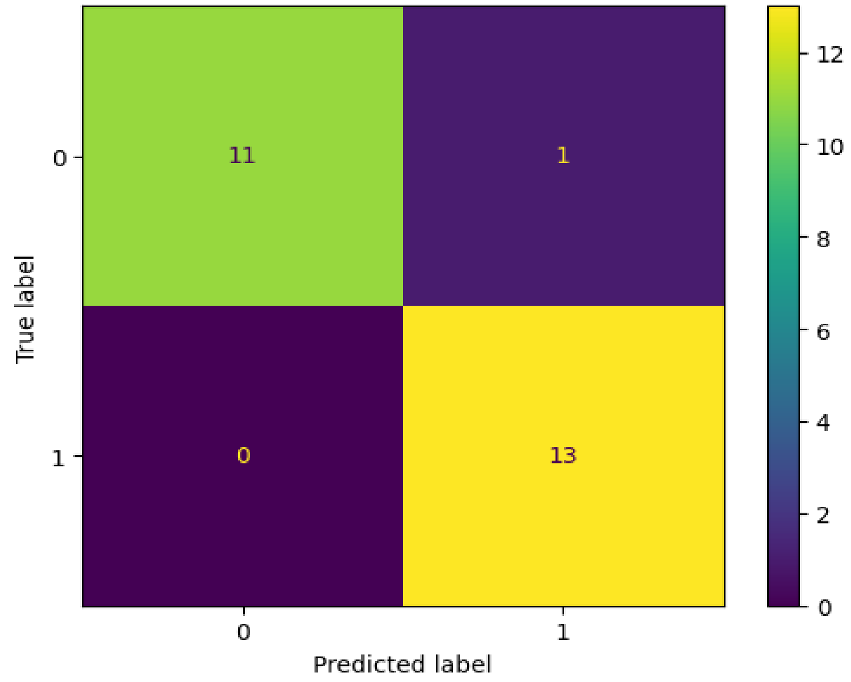
```python
test_df = pd.DataFrame(x_test, columns=['first', 'second'])
test_df['res'] = y_test
test_df.plot.scatter(x='first', y='second', c='res', colormap='viridis')
```

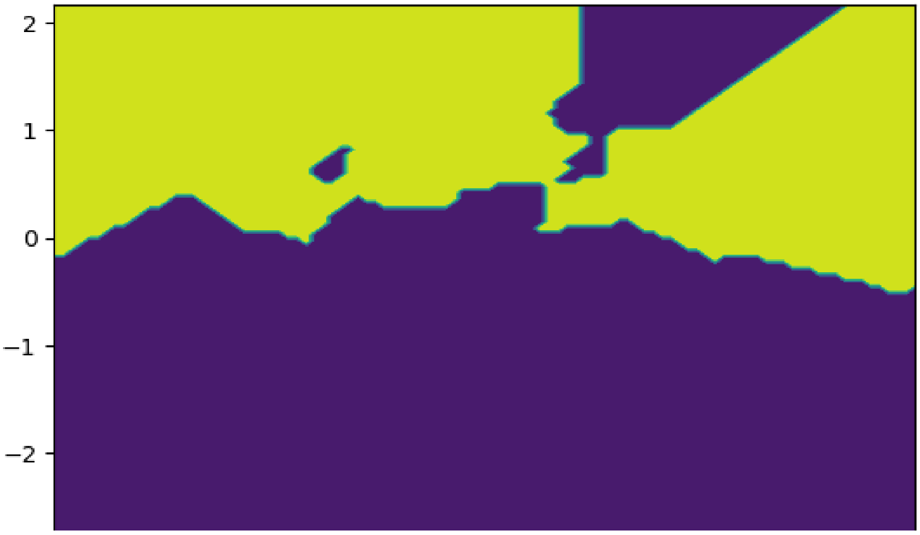
```
<Axes: xlabel='first', ylabel='second'>
```



```
def test_KNeighthboursClassifier_hyper(hyperparams):
    for param in hyperparams:
        print(f"param = {param}")
        clf = KNeighborsClassifier(n_neighbors=param)
        show_statistic(clf, x_test, y_test)

test_KNeighthboursClassifier_hyper([1])
```

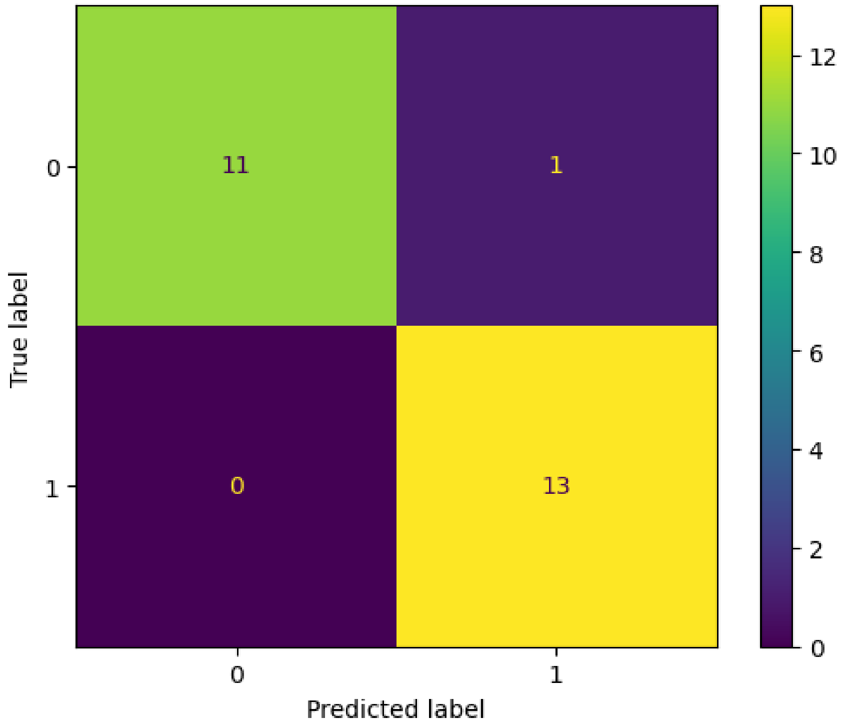
```
param = 1
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
```

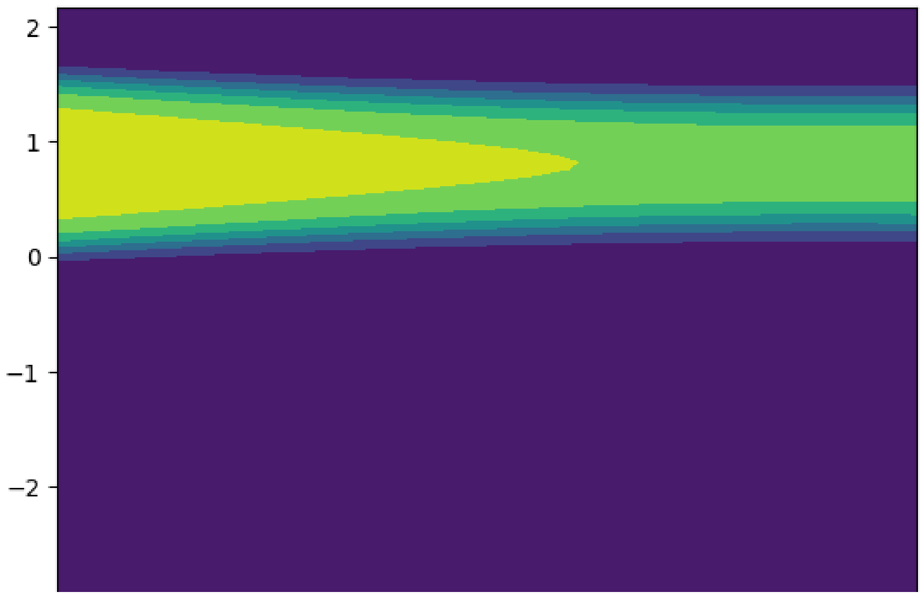




```
from sklearn.naive_bayes import GaussianNB

clf = GaussianNB()
clf.fit(x_train, y_train)
show_statistic(clf, x_test, y_test)
```

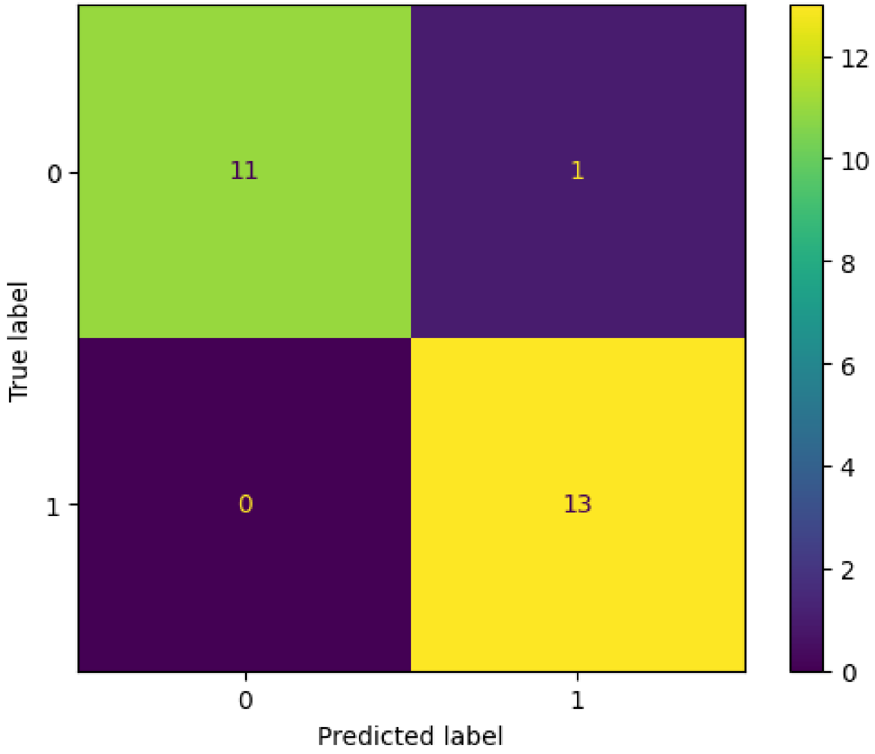
```
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
```

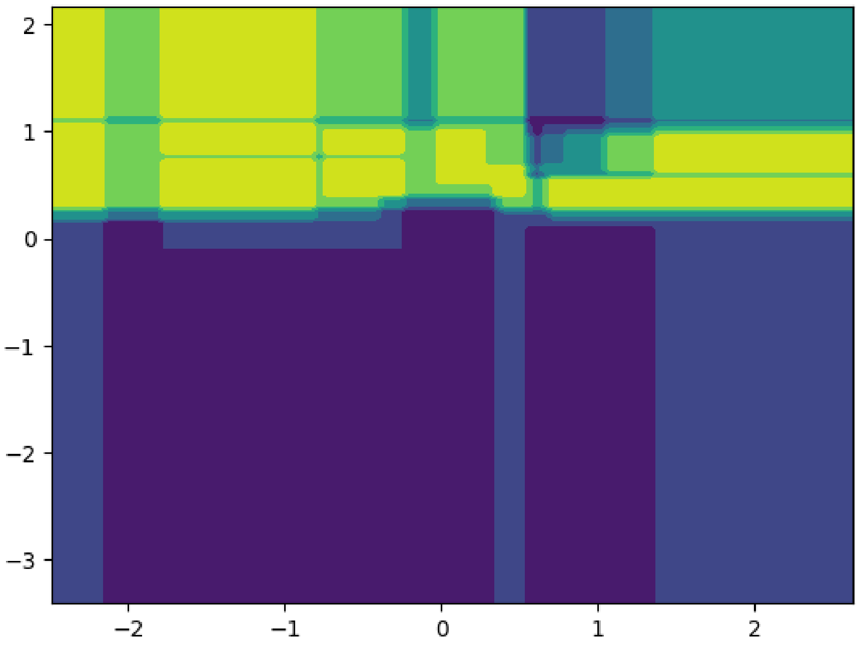


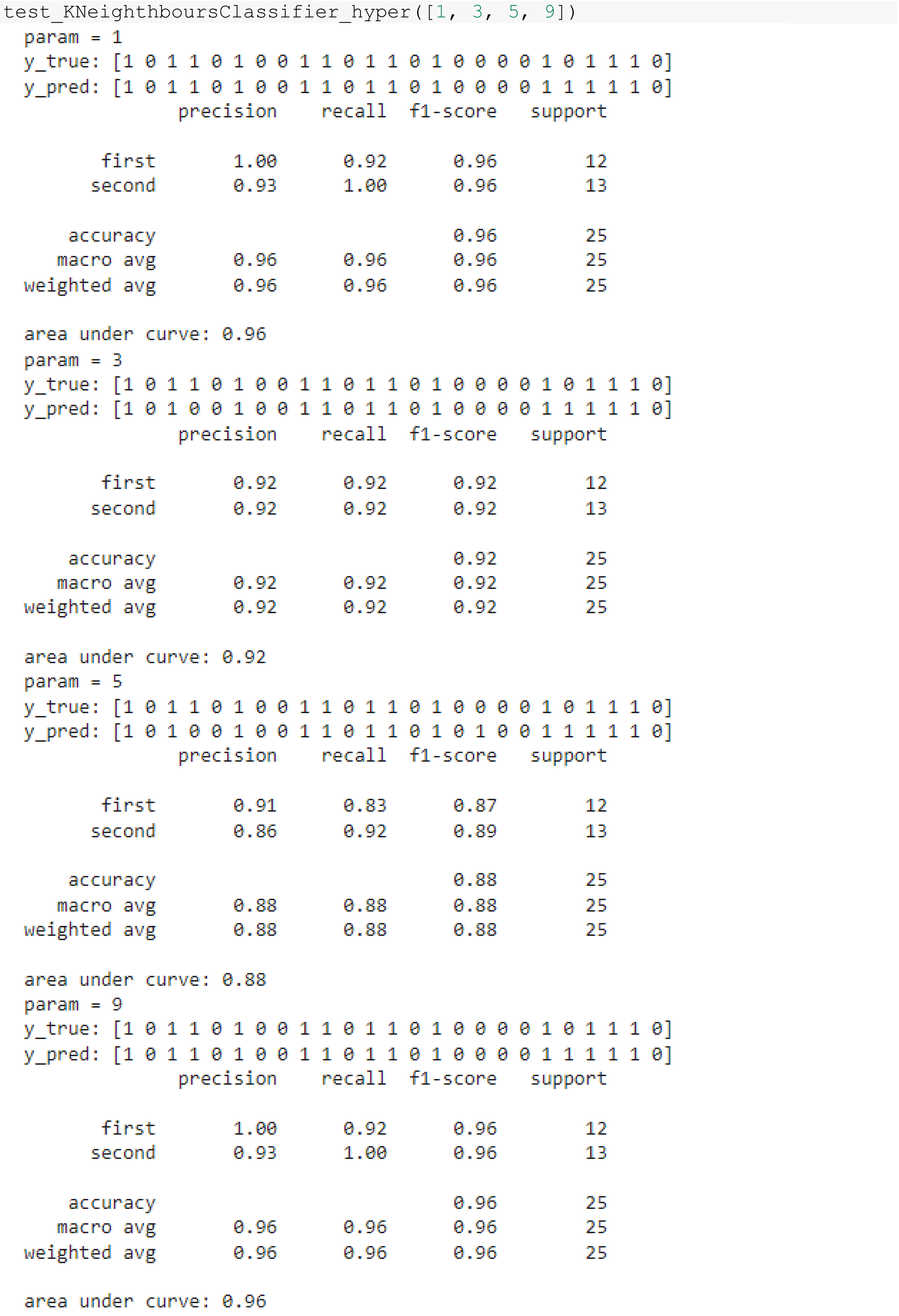


```python
from sklearn.ensemble import RandomForestClassifier

def test_RandomForestClassifier_hyper(hyperparams):
    for param in hyperparams:
        print(f"param = {param}")
        clf = RandomForestClassifier(n_estimators=param)
        clf.fit(x_train, y_train)
        show_statistic(clf, x_test, y_test)

test_RandomForestClassifier_hyper([5])
```

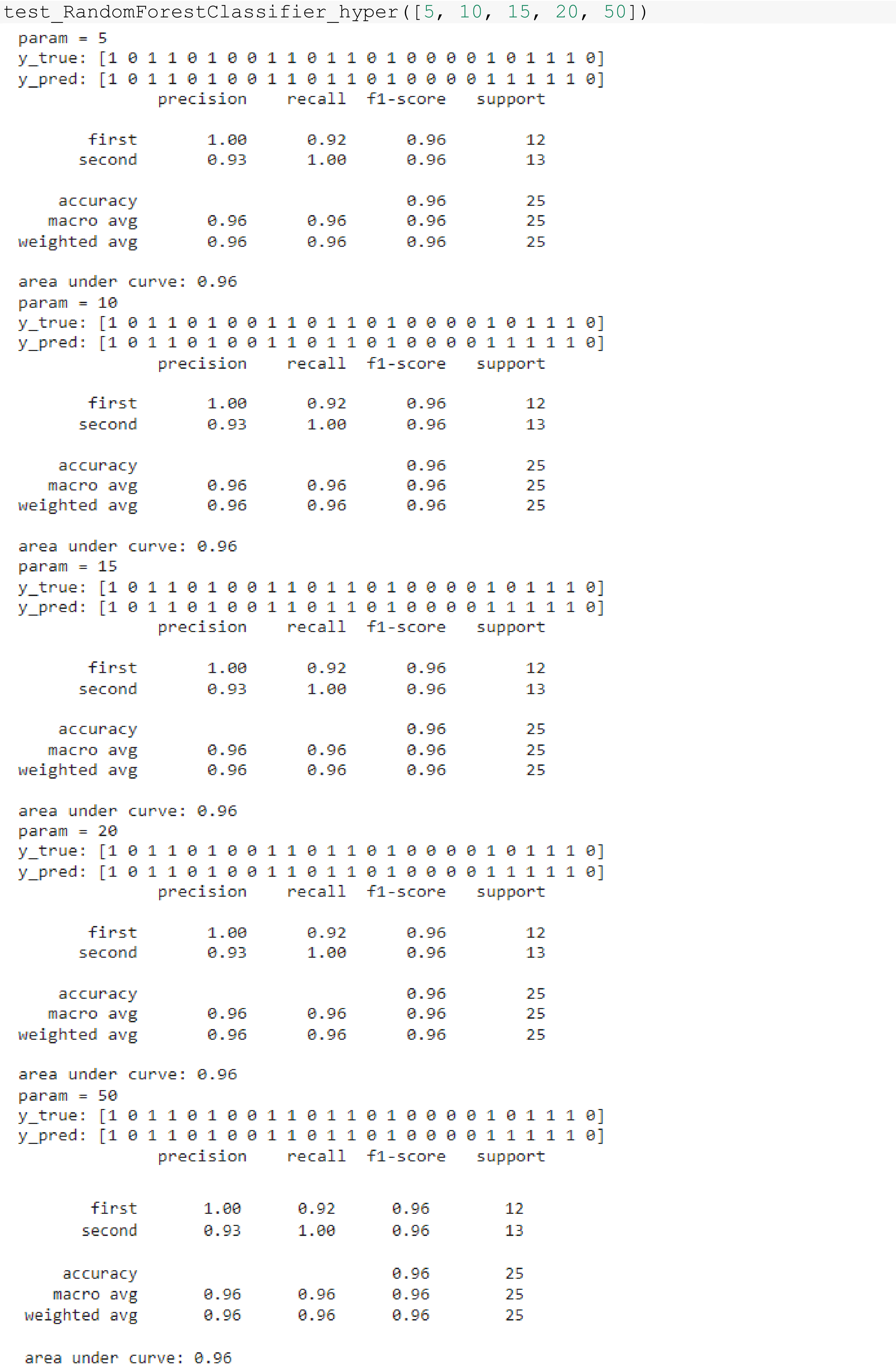
```
param = 5
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
```







```
test_KNeighthboursClassifier_hyper([1, 3, 5, 9])
  param = 1
  y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
  y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
                precision    recall  f1-score   support

         first       1.00      0.92      0.96        12
        second       0.93      1.00      0.96        13

      accuracy                           0.96        25
     macro avg       0.96      0.96      0.96        25
  weighted avg       0.96      0.96      0.96        25

  area under curve: 0.96
  param = 3
  y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
  y_pred: [1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
                precision    recall  f1-score   support

         first       0.92      0.92      0.92        12
        second       0.92      0.92      0.92        13

      accuracy                           0.92        25
     macro avg       0.92      0.92      0.92        25
  weighted avg       0.92      0.92      0.92        25

  area under curve: 0.92
  param = 5
  y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
  y_pred: [1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0 1 1 1 1 1 0]
                precision    recall  f1-score   support

         first       0.91      0.83      0.87        12
        second       0.86      0.92      0.89        13

      accuracy                           0.88        25
     macro avg       0.88      0.88      0.88        25
  weighted avg       0.88      0.88      0.88        25

  area under curve: 0.88
  param = 9
  y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
  y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
                precision    recall  f1-score   support

         first       1.00      0.92      0.96        12
        second       0.93      1.00      0.96        13

      accuracy                           0.96        25
     macro avg       0.96      0.96      0.96        25
  weighted avg       0.96      0.96      0.96        25

  area under curve: 0.96
```



```
test_RandomForestClassifier_hyper([5, 10, 15, 20, 50])
```

```
param = 5
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
param = 10
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
param = 15
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
param = 20
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
param = 50
y_true: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0]
y_pred: [1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0]
              precision    recall  f1-score   support

       first       1.00      0.92      0.96        12
      second       0.93      1.00      0.96        13

    accuracy                           0.96        25
   macro avg       0.96      0.96      0.96        25
weighted avg       0.96      0.96      0.96        25

area under curve: 0.96
```