**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

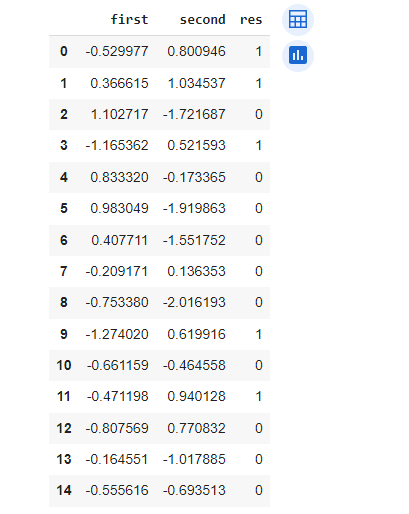
ds = make\_classification(random\_state=58, n\_informative=1, n\_redundant=0, n\_features=2, n\_clusters\_per\_class=1,

                         class\_sep=0.7)

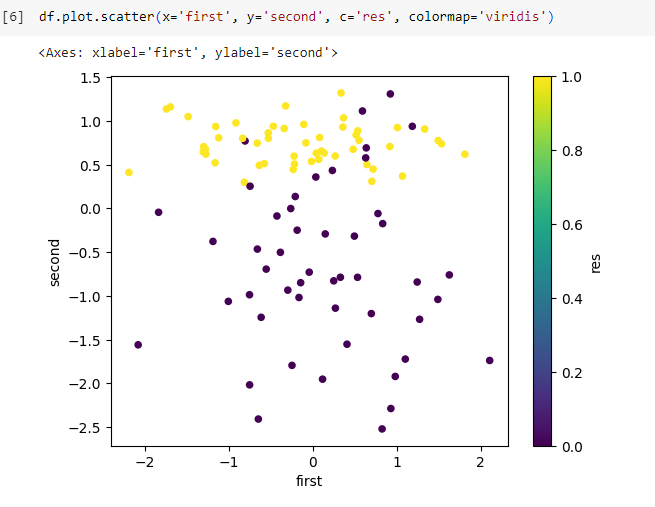
df = pd.DataFrame(ds[0], columns=['first', 'second'])

df['res'] = ds[1]

df.head(15)



Построим график, отображающий нашу выборку



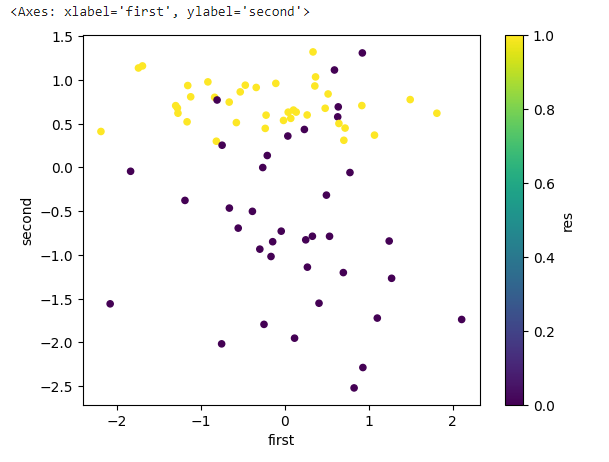
from sklearn.model\_selection import train\_test\_split

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

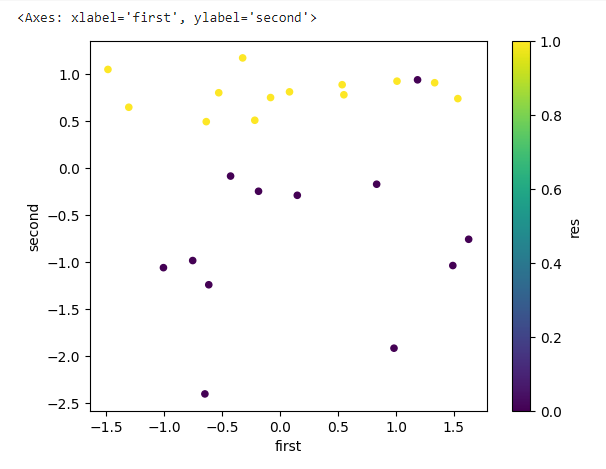
train\_df.plot.scatter(x='first', y='second', c='res', colormap='viridis')



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')



def test\_KNeighthboursClassifier\_hyper(hyperparams):

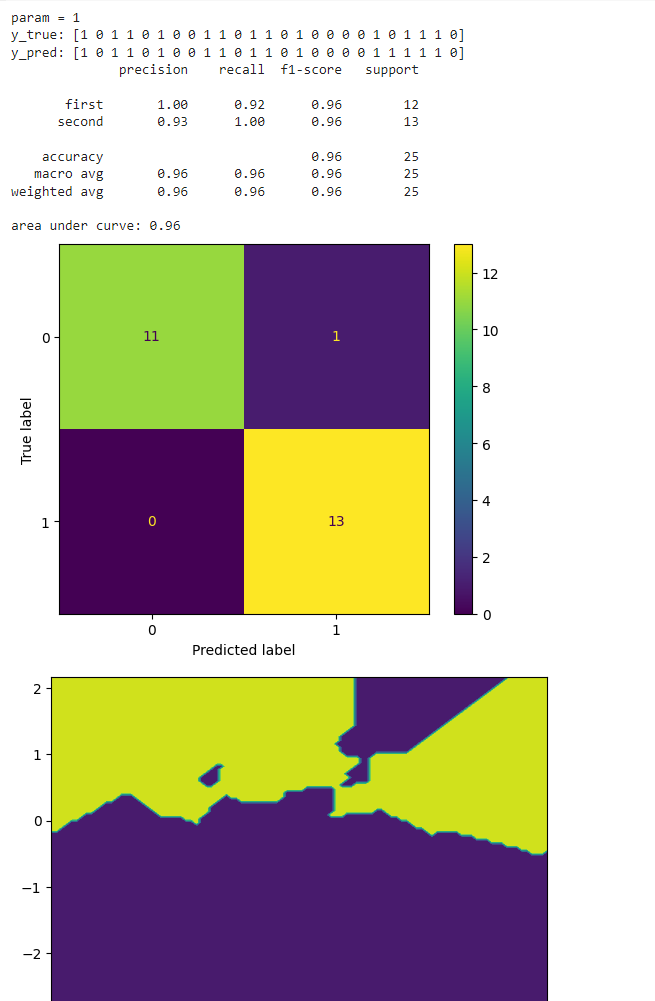
    for param in hyperparams:

        print(f"param = {param}")

        clf = KNeighborsClassifier(n\_neighbors=param)

        show\_statistic(clf, x\_test, y\_test)

test\_KNeighthboursClassifier\_hyper([1])

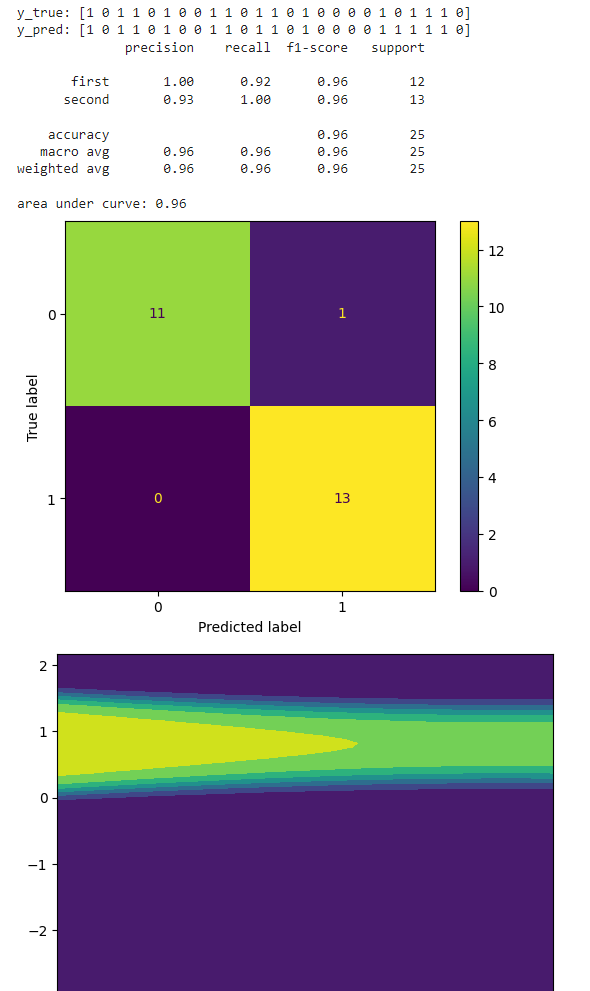


from sklearn.naive\_bayes import GaussianNB

clf = GaussianNB()

clf.fit(x\_train, y\_train)

show\_statistic(clf, x\_test, y\_test)



from sklearn.ensemble import RandomForestClassifier

def test\_RandomForestClassifier\_hyper(hyperparams):

    for param in hyperparams:

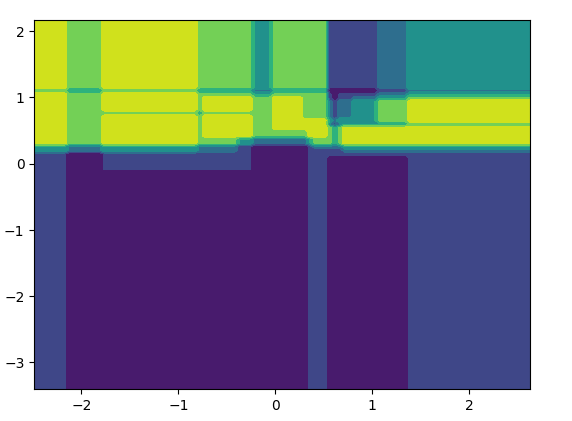
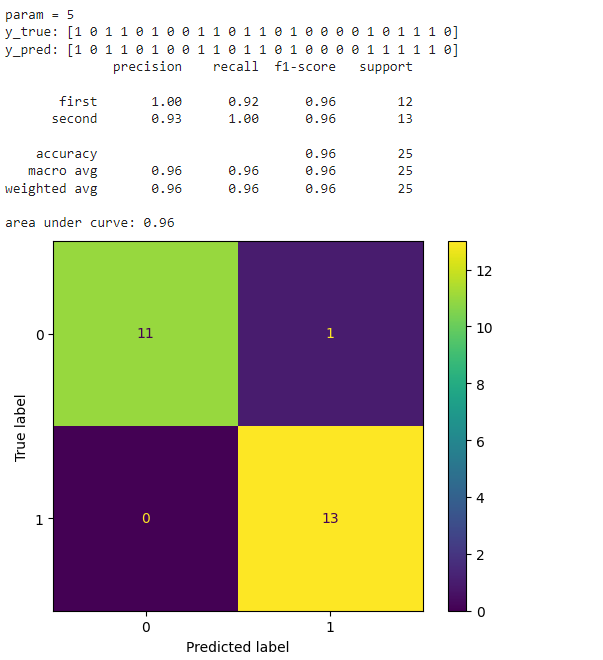
        print(f"param = {param}")

        clf = RandomForestClassifier(n\_estimators=param)

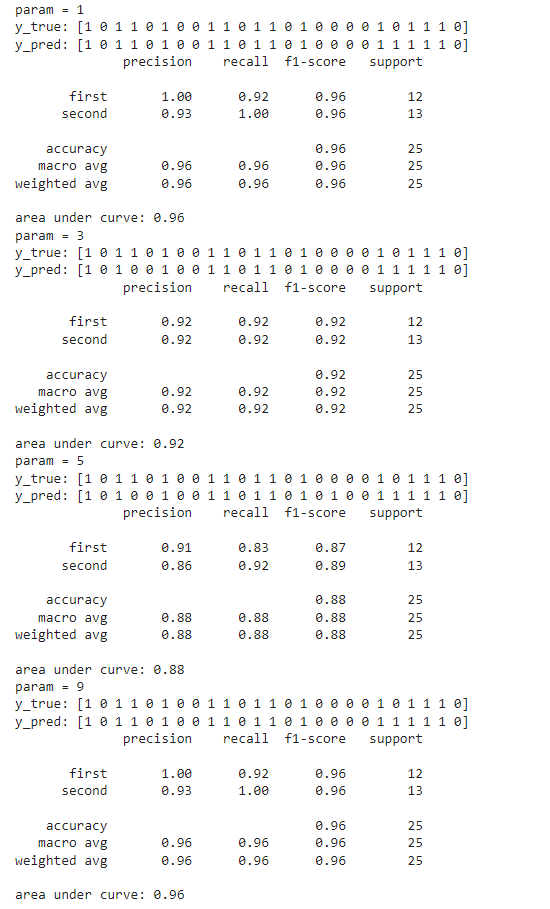
        clf.fit(x\_train, y\_train)

        show\_statistic(clf, x\_test, y\_test)

test\_RandomForestClassifier\_hyper([5])



test\_KNeighthboursClassifier\_hyper([1, 3, 5, 9])



test\_RandomForestClassifier\_hyper([5, 10, 15, 20, 50])

