

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №2

**по дисциплине «Прикладные интеллектуальные системы и экспертные
системы»**

Бинарная классификация

Студент

Крутских А.Ю.

Группа М-ИАП-22

Руководитель

Кургасов В.В.

Липецк 2022 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.

Задание кафедры

- 1) В среде Jupiter Notebook создать новый ноутбук (Notebook);
- 2) Импортировать необходимые для работы библиотеки и модули;
- 3) Загрузить данные в соответствие с вариантом;
- 4) Вывести первые 15 элементов выборки (координаты точек и метки класса);
- 5) Отобразить на графике сгенерированную выборку. Объекты разных классов должны иметь разные цвета;
- 6) Разбить данные на обучающую (train) и тестовую (test) выборки в пропорции 75% - 25% соответственно;
- 7) Отобразить на графике обучающую и тестовую выборки. Объекты разных классов должны иметь разные цвета;
- 8) Реализовать модели классификаторов, обучить их на обучающем множестве. Применить модели на тестовой выборке, вывести результаты классификации:
 - Истинные и предсказанные метки классов
 - Матрицу ошибок (confusion matrix)
 - Значения полноты, точности, f1-меры и аккуратности
 - Значение площади под кривой ошибок (AUC ROC)
 - Отобразить на графике область принятия решений по каждому классуВ качестве методов классификации использовать:
 - a) Метод k-ближайших соседей ($n_neighbors = \{1, 3, 5, 9\}$)
 - b) Наивный байесовский метод
 - c) Случайный лес ($n_estimators = \{5, 10, 15, 20, 50\}$)
- 9) По каждому пункту работы занести в отчет программный код и результат вывода;

10) По результатам п.8 занести в отчет таблицу с результатами классификации всеми методами и выводы о наиболее подходящем методе классификации ваших данных.

Ход работы

Вариант 5.

Вариант	5
Вид классов	moons
Random_state	41
cluster_std	-
noise	0.25
Centers	-

Для всех вариантов, использующих для генерации `make_classification`, дополнительные параметры: `n_features=2`, `n_redundant=0`, `n_informative=1`, `n_clusters_per_class=1`.

- 1) в среде Jupiter Notebook создать новый ноутбук (Notebook);
- 2) импортировать необходимые для работы библиотеки и модули;
- 3) загрузить данные в соответствии с вариантом;
- 4) вывести первые 15 элементов выборки (координаты точек и метки класса);
- 5) отобразить на графике сгенерированную выборку. Объекты разных классов должны иметь разные цвета;
- 6) разбить данные на обучающую (train) и тестовую (test) выборки в пропорции 75% - 25% соответственно;
- 7) отобразить на графике обучающую и тестовую выборки. Объекты разных классов должны иметь разные цвета;
- 8) реализовать модели классификаторов, обучить их на обучающем множестве. Применить модели на тестовой выборке, вывести результаты классификации:
 - Истинные и предсказанные метки классов
 - Матрицу ошибок (confusion matrix)
 - Значения полноты, точности, f1-меры и аккуратности

- Значение площади под кривой ошибок (AUC ROC)
 - Отобразить на графике область принятия решений по каждому классу
- В качестве методов классификации использовать:
- a) Метод k-ближайших соседей ($n_neighbors = \{1, 3, 5, 9\}$)
 - b) Наивный байесовский метод
 - c) Случайный лес ($n_estimators = \{5, 10, 15, 20, 50\}$)
- 9) по каждому пункту работы занести в отчет программный код и результат вывода;
- 10) по результатам п.8 занести в отчет таблицу с результатами классификации всеми методами и выводы о наиболее подходящем методе классификации ваших данных.

Ход работы

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
```

Рисунок 1 - Импортируем библиотеки

Вид класса	random_state	cluster_std	noise	centers
blobs	41	3	-	2

```
x, y = make_moons(noise=0.25, random_state=41)

print("Координаты x y:\n", x[:15])
print("Метки класса: ", y[:15])

Координаты x y:
[[-1.53415216 -0.08217918]
 [ 2.16440543 -0.2734159 ]
 [-0.58527459  0.85376957]
 [ 1.35195628  0.38893433]
 [ 1.46479345 -0.25172895]
 [-0.64362199  1.06018572]
 [ 0.40847955  0.87854153]
 [ 1.16692891 -0.03600687]
 [-0.70389555  0.90891989]
 [ 1.69466082 -0.36203457]
 [ 0.60290958 -0.69829682]
 [ 0.16150261 -0.15701266]
 [-0.07098503 -0.50788692]
 [ 0.6133523  0.04294663]
 [ 0.43329615  0.04721913]]
Метки класса: [0 1 0 0 1 0 0 0 0 1 1 1 1 0 1]
```

Рисунок 2 – Генерируем выборку

```
plt.scatter(x[:,0], x[:,1], c=y)

<matplotlib.collections.PathCollection at 0x1e6b6091030>
```

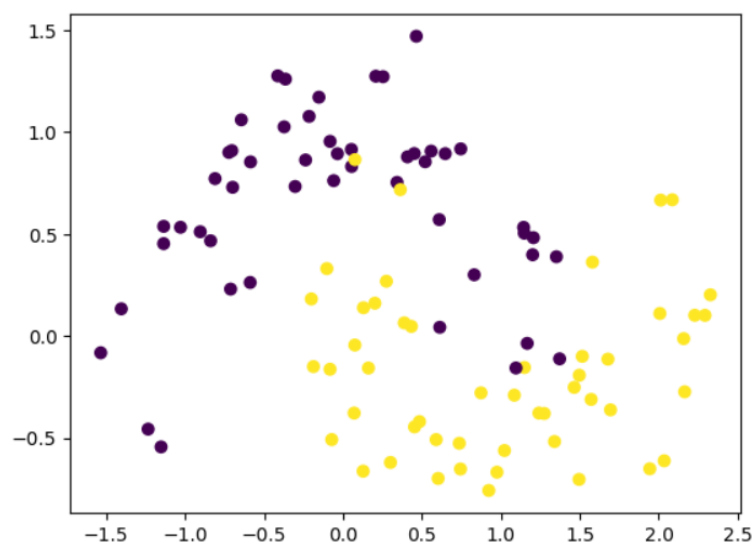


Рисунок 3 – График сгенерированной выборки

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state=1)
```

Рисунок 4 - Разбиваем данные на обучающую (train) и тестовую (test) выборки в пропорции 75% - 25% соответственно

```
plt.scatter(x_train[:,0], x_train[:,1], c=y_train)  
<matplotlib.collections.PathCollection at 0x1e6b8416980>
```

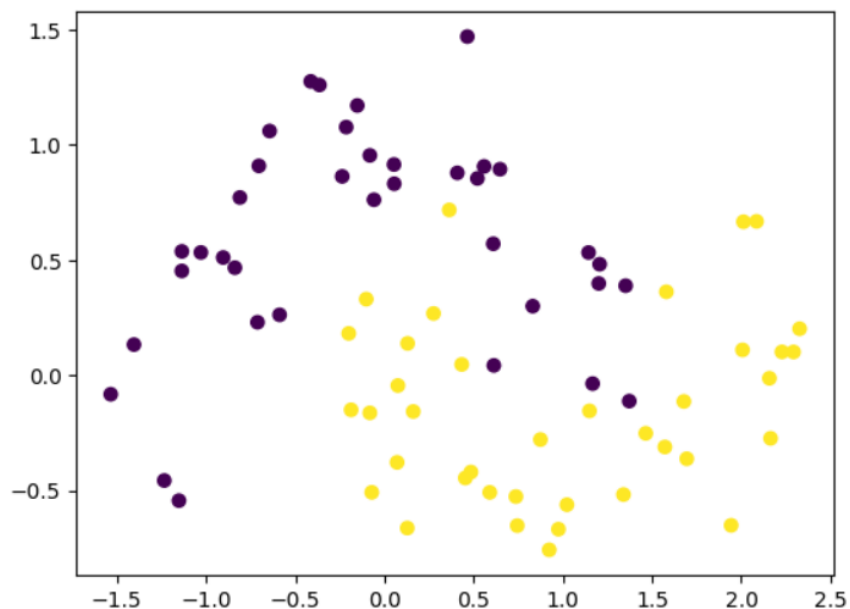


Рисунок 5 - График обучающей выборки

```
In [8]: plt.scatter(x_test[:,0], x_test[:,1], c=y_test)
```



Рисунок 6 - График тестовой выборки


```

for i in [1, 3, 5, 9]:
    knn = KNeighborsClassifier(n_neighbors=i, metric='euclidean')
    knn.fit(x_train, y_train)
    prediction = knn.predict(x_test)
    print("n_neighbors = ", i)
    print_classification_metrics(knn, x, y, prediction, y_test)

```

n_neighbors = 1

Предсказанные и истинные значения

[1 1 1 0 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 0 0 1 0]

[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]

Матрица ошибок

[[11 2]

[2 10]]

Точность классификации: 0.84

Значения полноты, точности, f1-меры и аккуратности

	precision	recall	f1-score	support
0	0.85	0.85	0.85	13
1	0.83	0.83	0.83	12
accuracy			0.84	25
macro avg	0.84	0.84	0.84	25
weighted avg	0.84	0.84	0.84	25

Значение площади под кривой ошибок (AUC ROC)

0.8397435897435898

Значение площади под кривой ошибок (AUC ROC)

0.8397435897435898

Область принятия решений

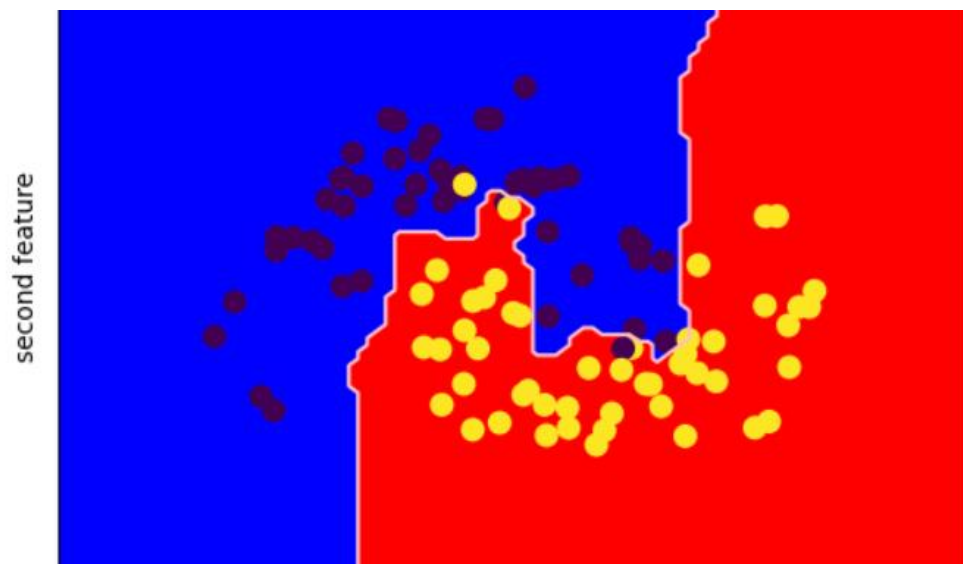


Рисунок 7 – Метод k-ближайших соседей (n=1)

```

n_neighbors = 3
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
Матрица ошибок
[[12  1]
 [ 1 11]]
Точность классификации: 0.92
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0       0.92       0.92       0.92        13
      1       0.92       0.92       0.92        12

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

Значение площади под кривой ошибок (AUC ROC)
0.9198717948717948
Область принятия решений

```

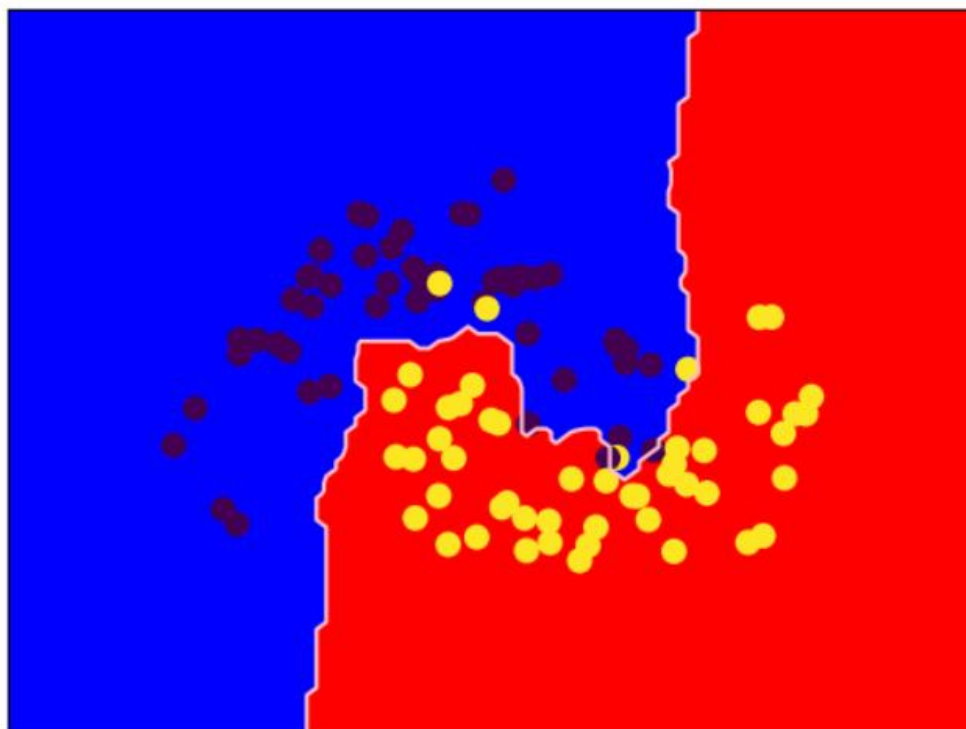


Рисунок 8 – Метод k-ближайших соседей (n=3)

```

n_neighbors = 5
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
Матрица ошибок
[[12  1]
 [ 1 11]]
Точность классификации: 0.92
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0         0.92      0.92      0.92        13
      1         0.92      0.92      0.92        12

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

Значение площади под кривой ошибок (AUC ROC)
0.9198717948717948
Область принятия решений

```

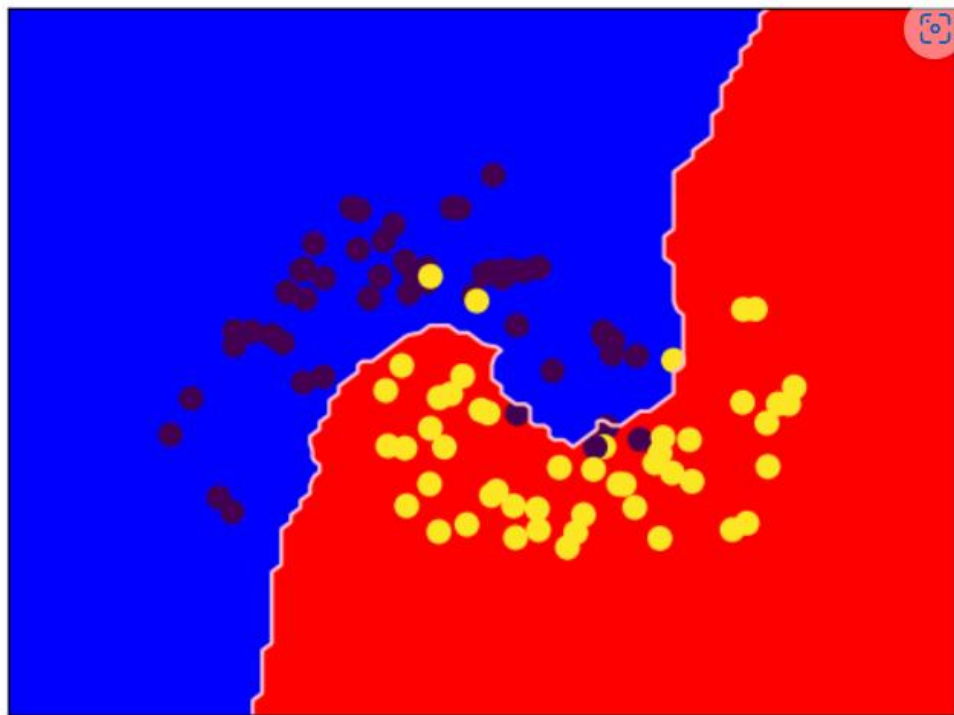


Рисунок 9 – Метод k-ближайших соседей (n=5)

```

n_neighbors = 9
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
Матрица ошибок
[[12  1]
 [ 1 11]]
Точность классификации: 0.92
Значения полноты, точности, f1-меры и аккуратности

```

	precision	recall	f1-score	support
0	0.92	0.92	0.92	13
1	0.92	0.92	0.92	12
accuracy			0.92	25
macro avg	0.92	0.92	0.92	25
weighted avg	0.92	0.92	0.92	25

```

Значение площади под кривой ошибок (AUC ROC)
0.9198717948717948
Область принятия решений

```

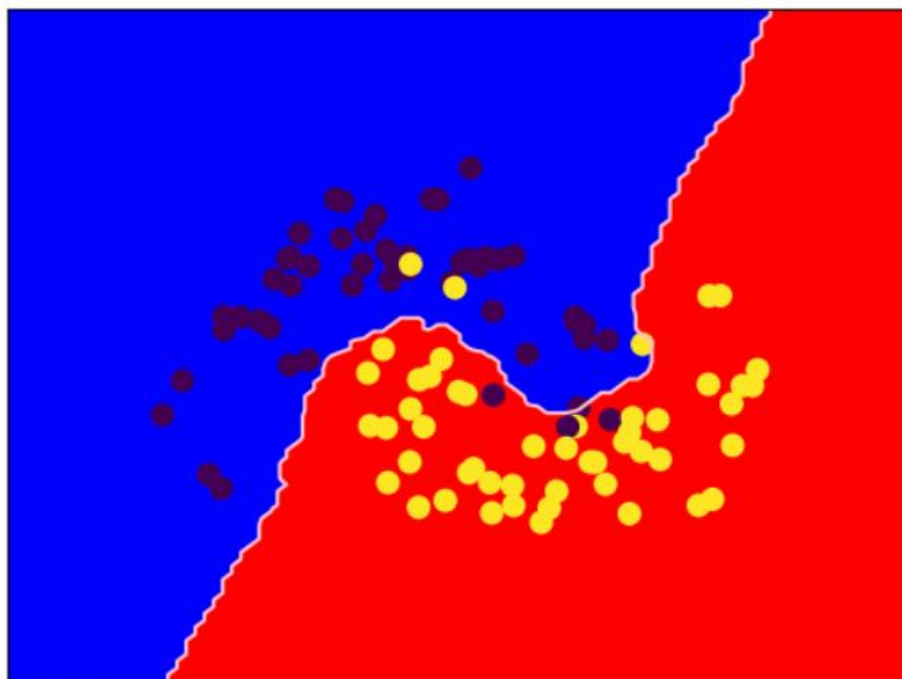


Рисунок 10 – Метод k-ближайших соседей (n=9)

```
naive = GaussianNB()
naive.fit(x_train, y_train)
predict = naive.predict(x_test)
print_classification_metrics(naive, x, y, predict, y_test)
```

Предсказанные и истинные значения

```
[0 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
```

Матрица ошибок

```
[[12  1]
 [ 2 10]]
```

Точность классификации: 0.88

Значения полноты, точности, f1-меры и аккуратности

	precision	recall	f1-score	support
0	0.86	0.92	0.89	13
1	0.91	0.83	0.87	12
accuracy			0.88	25
macro avg	0.88	0.88	0.88	25
weighted avg	0.88	0.88	0.88	25

Значение площади под кривой ошибок (AUC ROC)

0.8782051282051283

Область принятия решений

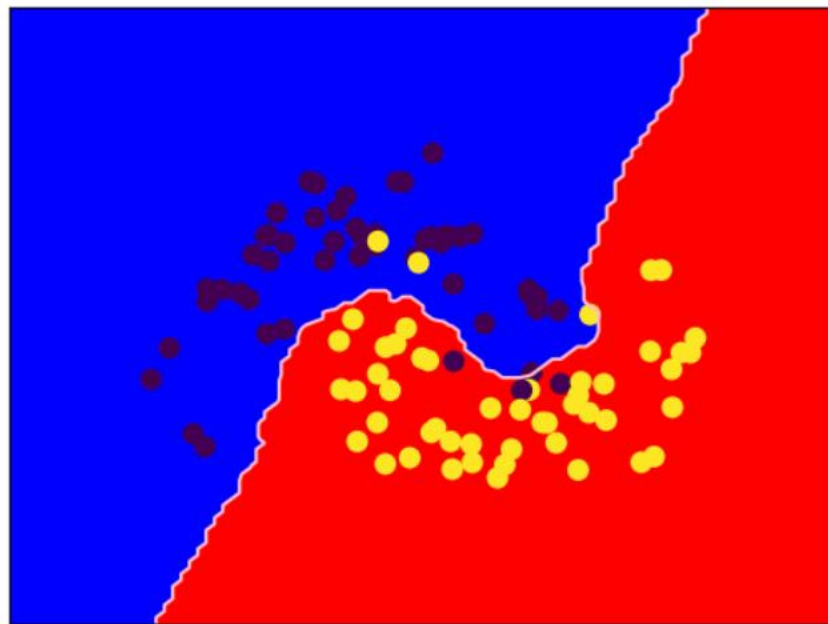


Рисунок 11 – Наивный байесовский метод

```

for i in [5, 10, 15, 20, 50]:
    rand_forest = RandomForestClassifier(n_estimators=i)
    rand_forest.fit(x_train, y_train)
    prediction = rand_forest.predict(x_test)
    print("n_estimators = ", i)
    print_classification_metrics(knn, x, y, prediction, y_test)

```

n_estimators = 5

Предсказанные и истинные значения

[1 1 1 0 0 0 1 1 1 0 1 1 0 1 0 0 1 0 0 0 1 0]

[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0]

Матрица ошибок

```

[[11  2]
 [ 1 11]]

```

Точность классификации: 0.88

Значения полноты, точности, f1-меры и аккуратности

	precision	recall	f1-score	support
0	0.92	0.85	0.88	13
1	0.85	0.92	0.88	12
accuracy			0.88	25
macro avg	0.88	0.88	0.88	25
weighted avg	0.88	0.88	0.88	25

Значение площади под кривой ошибок (AUC ROC)

0.8814102564102563

Область принятия решений

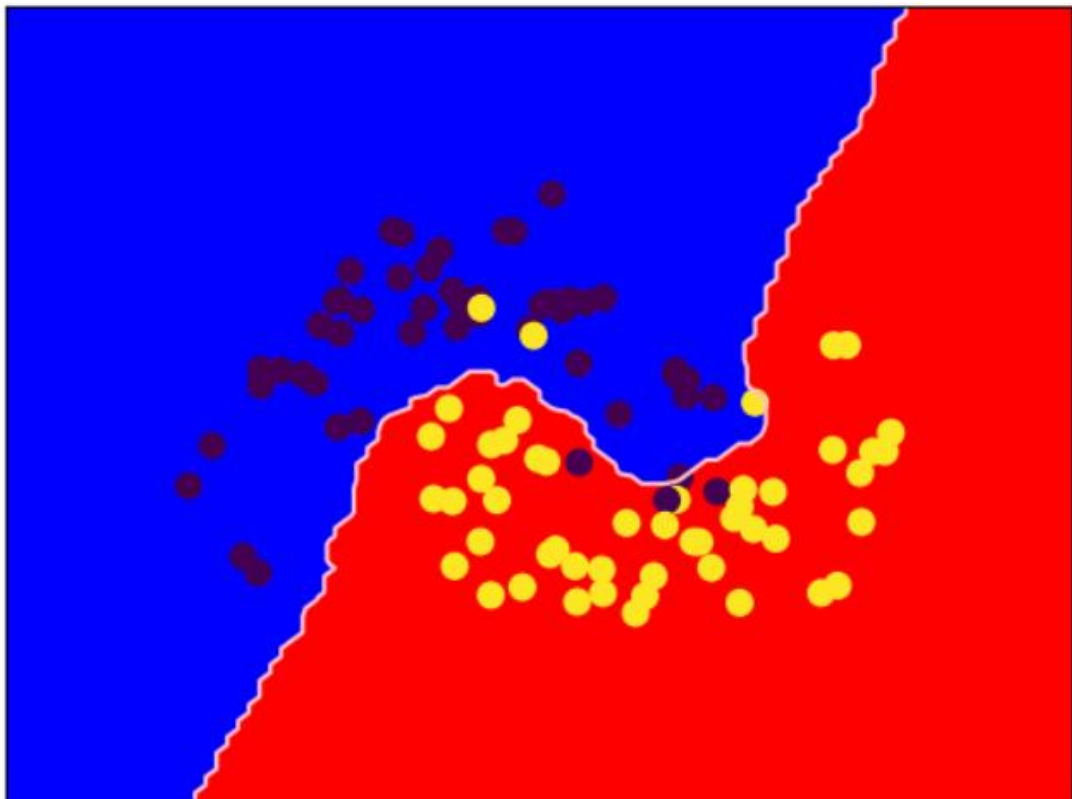


Рисунок 12 – Случайный лес n=5

```

n_estimators = 10
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0]
Матрица ошибок
[[12  1]
 [ 1 11]]
Точность классификации: 0.92
Значения полноты, точности, f1-меры и аккуратности

```

	precision	recall	f1-score	support
0	0.92	0.92	0.92	13
1	0.92	0.92	0.92	12
accuracy			0.92	25
macro avg	0.92	0.92	0.92	25
weighted avg	0.92	0.92	0.92	25

```

Значение площади под кривой ошибок (AUC ROC)
0.9198717948717948
Область принятия решений

```

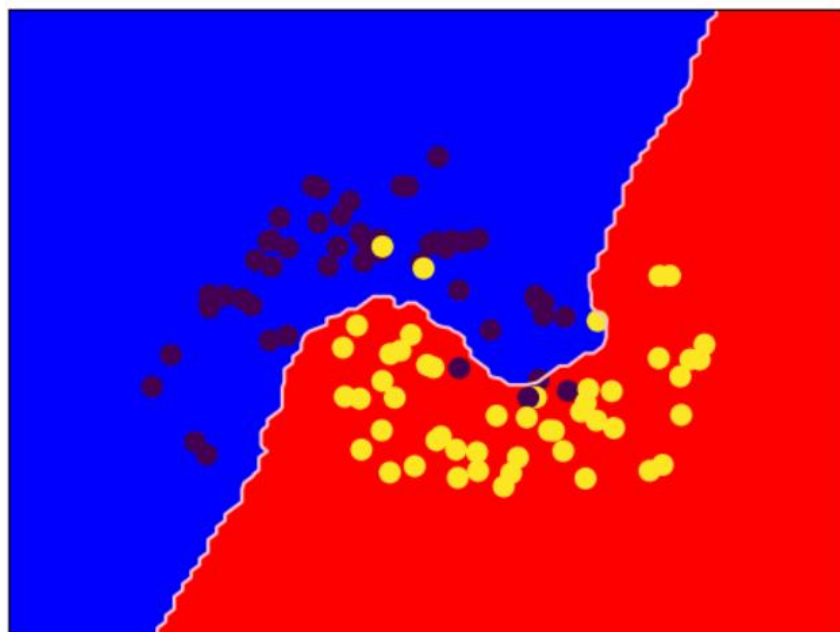


Рисунок 13 – Случайный лес n=10

```

n_estimators = 15
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
Матрица ошибок
[[12  1]
 [ 1 11]]
Точность классификации: 0.92
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0         0.92      0.92      0.92         13
     1         0.92      0.92      0.92         12

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

Значение площади под кривой ошибок (AUC ROC)
0.9198717948717948
Область принятия решений

```

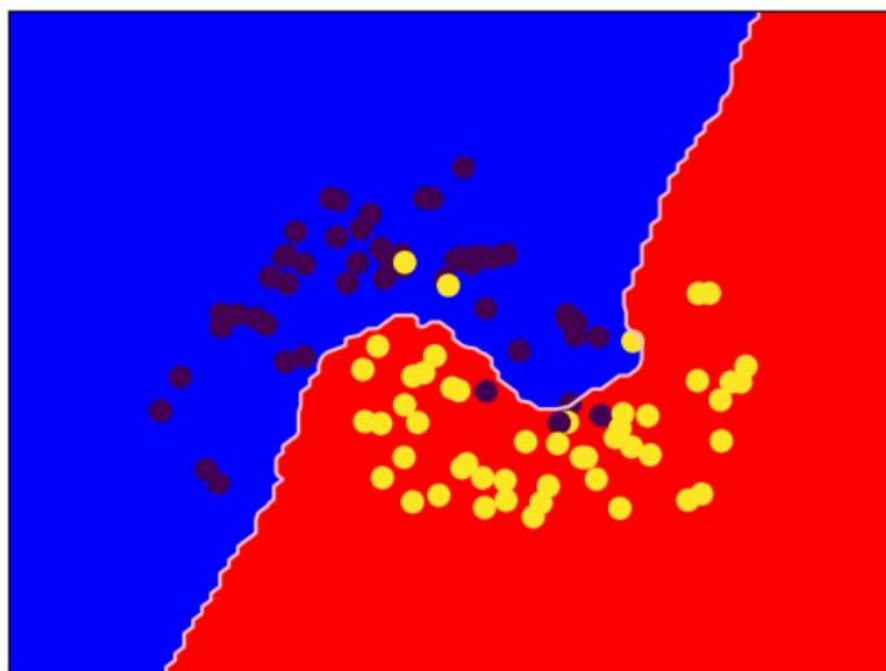


Рисунок 14 – Случайный лес n=15


```

n_estimators = 20
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 1 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
Матрица ошибок
[[11  2]
 [ 1 11]]
Точность классификации: 0.88
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.92       0.85       0.88        13
     1       0.85       0.92       0.88        12

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

Значение площади под кривой ошибок (AUC ROC)
0.8814102564102563
Область принятия решений

```

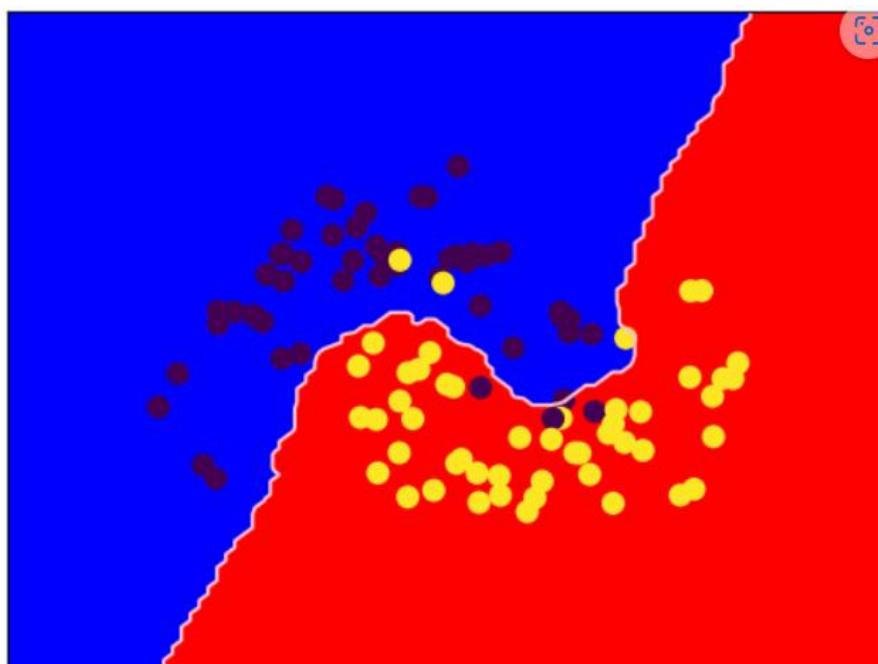


Рисунок 14 – Случайный лес n=20

```

n_estimators = 50
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
Матрица ошибок
[[12  1]
 [ 1 11]]
Точность классификации: 0.92
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0         0.92      0.92      0.92        13
      1         0.92      0.92      0.92        12

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

Значение площади под кривой ошибок (AUC ROC)
0.9198717948717948
Область принятия решений

```

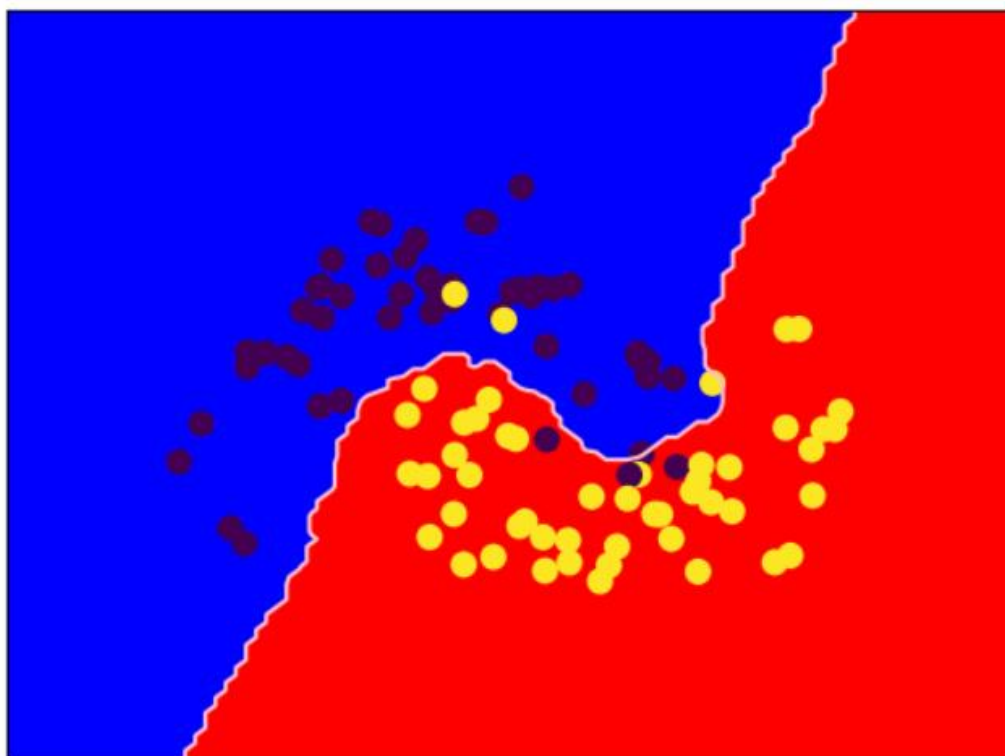


Рисунок 15 – Случайный лес n=50

Таблица 1 – Результаты работы программы

Метод	Истинные и предсказанные метки классов	Матрица ошибок	Значения полноты, точности, f1- меры и аккуратности				Значение площади под кривой ошибок
			precision	recall	f1-score	support	
			0	0.85	0.85	0.85	13
			1	0.83	0.83	0.83	12
			accuracy			0.84	25
			macro avg	0.84	0.84	0.84	25
			weighted avg	0.84	0.84	0.84	25
k-ближайших соседей n=1	[1 1 1 0 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0] [1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]	[[11 2] [2 10]]					0.8397435897435898
			Значения полноты, точности, f1-меры и аккуратности				
			precision	recall	f1-score	support	
			0	0.92	0.92	0.92	13
			1	0.92	0.92	0.92	12
			accuracy			0.92	25
			macro avg	0.92	0.92	0.92	25
			weighted avg	0.92	0.92	0.92	25
k-ближайших соседей n=3	[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0] [1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]	[[12 1] [1 11]]					0.9198717948717948
			Значения полноты, точности, f1-меры и аккуратности				
			precision	recall	f1-score	support	
			0	0.92	0.92	0.92	13
			1	0.92	0.92	0.92	12
			accuracy			0.92	25
			macro avg	0.92	0.92	0.92	25
			weighted avg	0.92	0.92	0.92	25
k-ближайших соседей n=5	[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0] [1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]	[[12 1] [1 11]]					0.9198717948717948
			Значения полноты, точности, f1-меры и аккуратности				
			precision	recall	f1-score	support	
			0	0.92	0.92	0.92	13
			1	0.92	0.92	0.92	12
			accuracy			0.92	25
			macro avg	0.92	0.92	0.92	25
			weighted avg	0.92	0.92	0.92	25
k-ближайших соседей n=9	[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0] [1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]	[[12 1] [1 11]]					0.9198717948717948
			Значения полноты, точности, f1-меры и аккуратности				
			precision	recall	f1-score	support	
			0	0.92	0.92	0.92	13
			1	0.92	0.92	0.92	12
			accuracy			0.92	25
			macro avg	0.92	0.92	0.92	25
			weighted avg	0.92	0.92	0.92	25

Наивный
байесовский
метод

```
[0 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
```

```
[[12 1]
 [ 2 10]]
```

Значения полноты, точности, f1-меры и аккуратности				
	precision	recall	f1-score	support
0	0.86	0.92	0.89	13
1	0.91	0.83	0.87	12
accuracy			0.88	25
macro avg	0.88	0.88	0.88	25
weighted avg	0.88	0.88	0.88	25

0.8782051282051283

Случайный лес
n=5

```
[1 1 1 0 0 0 1 1 1 0 1 1 0 1 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
```

```
[[11 2]
 [ 1 11]]
```

Значения полноты, точности, f1-меры и аккуратности				
	precision	recall	f1-score	support
0	0.92	0.85	0.88	13
1	0.85	0.92	0.88	12
accuracy			0.88	25
macro avg	0.88	0.88	0.88	25
weighted avg	0.88	0.88	0.88	25

0.8814102564102563

Случайный лес
n=10

```
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
```

```
[[12 1]
 [ 1 11]]
```

Значения полноты, точности, f1-меры и аккуратности				
	precision	recall	f1-score	support
0	0.92	0.92	0.92	13
1	0.92	0.92	0.92	12
accuracy			0.92	25
macro avg	0.92	0.92	0.92	25
weighted avg	0.92	0.92	0.92	25

0.9198717948717948

Случайный лес
n=15

```
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
```

```
[[12 1]
 [ 1 11]]
```

Значения полноты, точности, f1-меры и аккуратности				
	precision	recall	f1-score	support
0	0.92	0.92	0.92	13
1	0.92	0.92	0.92	12
accuracy			0.92	25
macro avg	0.92	0.92	0.92	25
weighted avg	0.92	0.92	0.92	25

0.9198717948717948

Случайный лес
n=20

```
[1 1 1 0 0 0 1 1 1 0 1 1 0 1 1 0 1 0 0 1 0 0 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0]
```

```
[[11 2]
 [ 1 11]]
```

Значения полноты, точности, f1-меры и аккуратности				
	precision	recall	f1-score	support
0	0.92	0.85	0.88	13
1	0.85	0.92	0.88	12
accuracy			0.88	25
macro avg	0.88	0.88	0.88	25
weighted avg	0.88	0.88	0.88	25

0.8814102564102563

```
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0] [[12 1]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0] [ 1 11]]
```

Значения полноты, точности, f1-меры и аккуратности				
	precision	recall	f1-score	support
0	0.92	0.92	0.92	13
1	0.92	0.92	0.92	12
accuracy			0.92	25
macro avg	0.92	0.92	0.92	25
weighted avg	0.92	0.92	0.92	25

0.9198717948717948

Случайный лес
n=50

Аккуратность при данном разбиении больше всего у метода случайного леса, следовательно он подходит для классификации данных.
Разобьем данные на обучающие (train) и тестовые (test) выборки в пропорции 90% - 10% соответственно.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.10, random_state=1)
```

Рисунок 16 – Разбиение данных на обучающие и тестовые

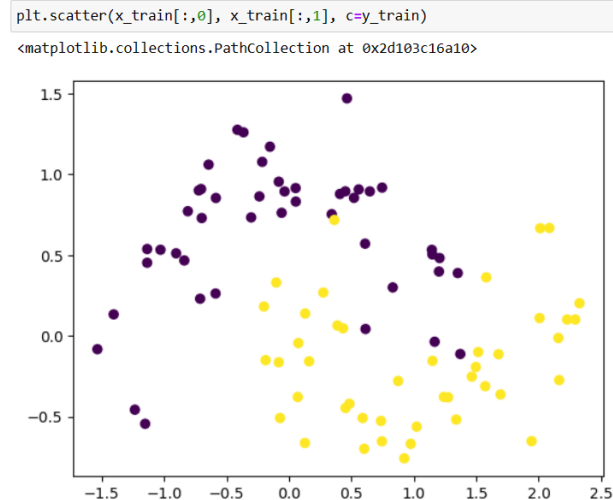


Рисунок 17 – График обучающей выборки

```
plt.scatter(x_test[:,0], x_test[:,1], c=y_test)
```

```
<matplotlib.collections.PathCollection at 0x2d103cad1e0>
```

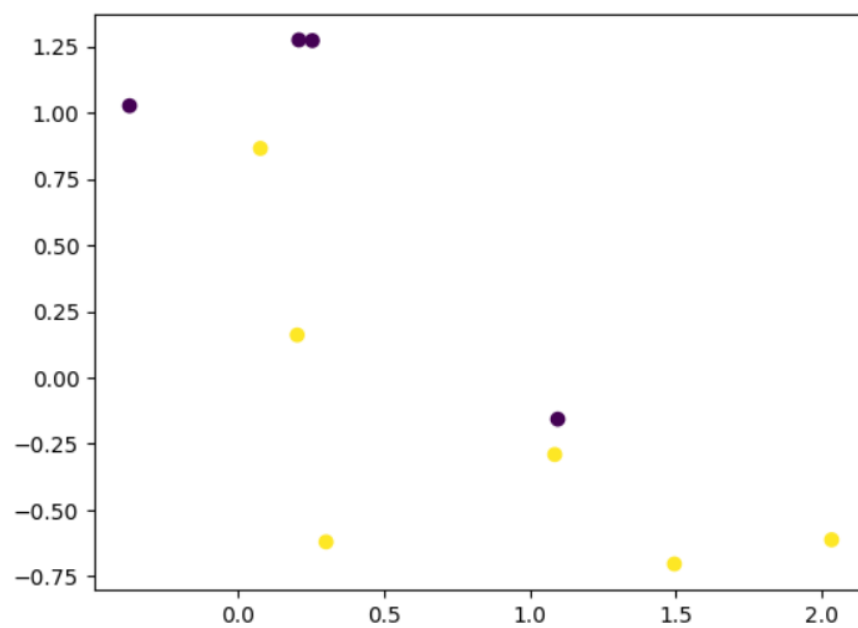


Рисунок 18 – График тестовой выборки

```

n_neighbors = 1
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.75      0.75      0.75         4
     1       0.83      0.83      0.83         6

 accuracy          0.80         10
 macro avg       0.79      0.79      0.79         10
weighted avg       0.80      0.80      0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

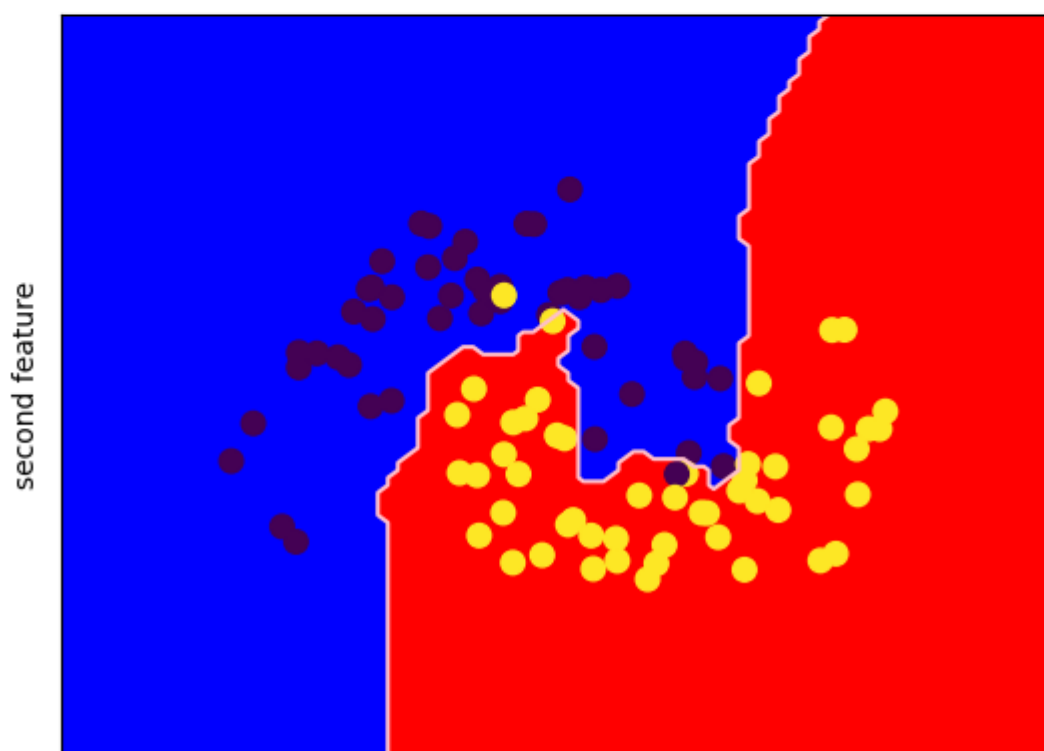


Рисунок 19 – Метод k-ближайших соседей n=1

```

n_neighbors = 3
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
precision    recall  f1-score   support

      0       0.75      0.75      0.75         4
      1       0.83      0.83      0.83         6

 accuracy          0.80         10
 macro avg       0.79      0.79      0.79         10
weighted avg       0.80      0.80      0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

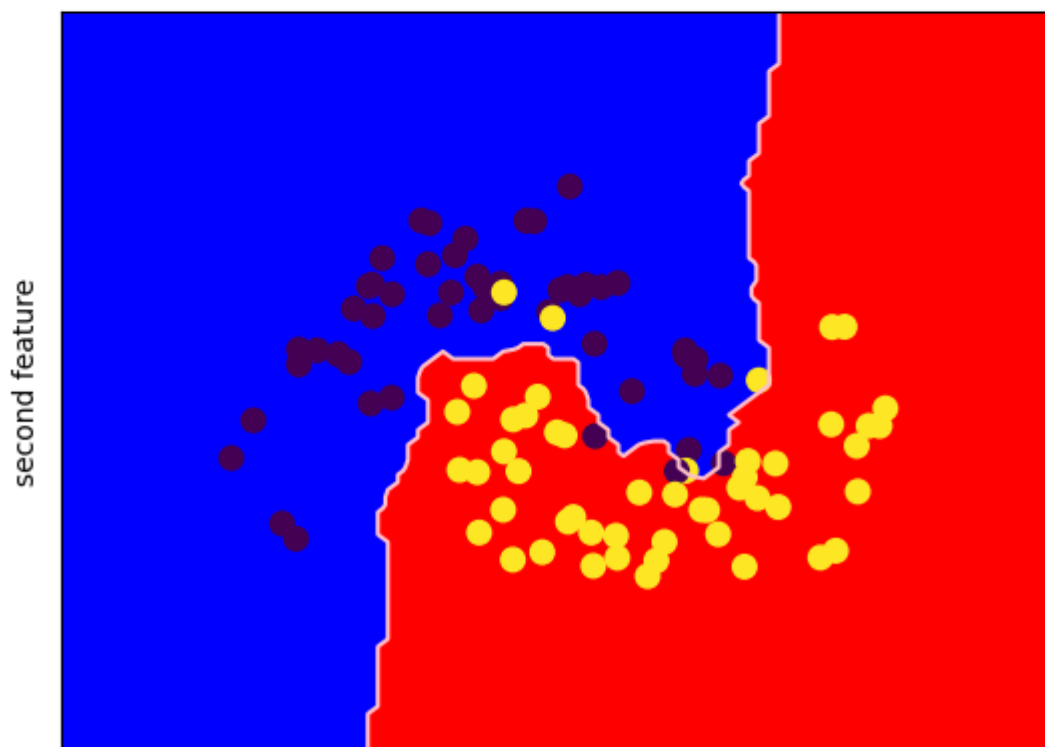


Рисунок 20 – Метод k-ближайших соседей n=3


```

n_neighbors = 5
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0       0.75      0.75      0.75         4
      1       0.83      0.83      0.83         6

 accuracy          0.80         10
 macro avg       0.79      0.79      0.79         10
weighted avg       0.80      0.80      0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

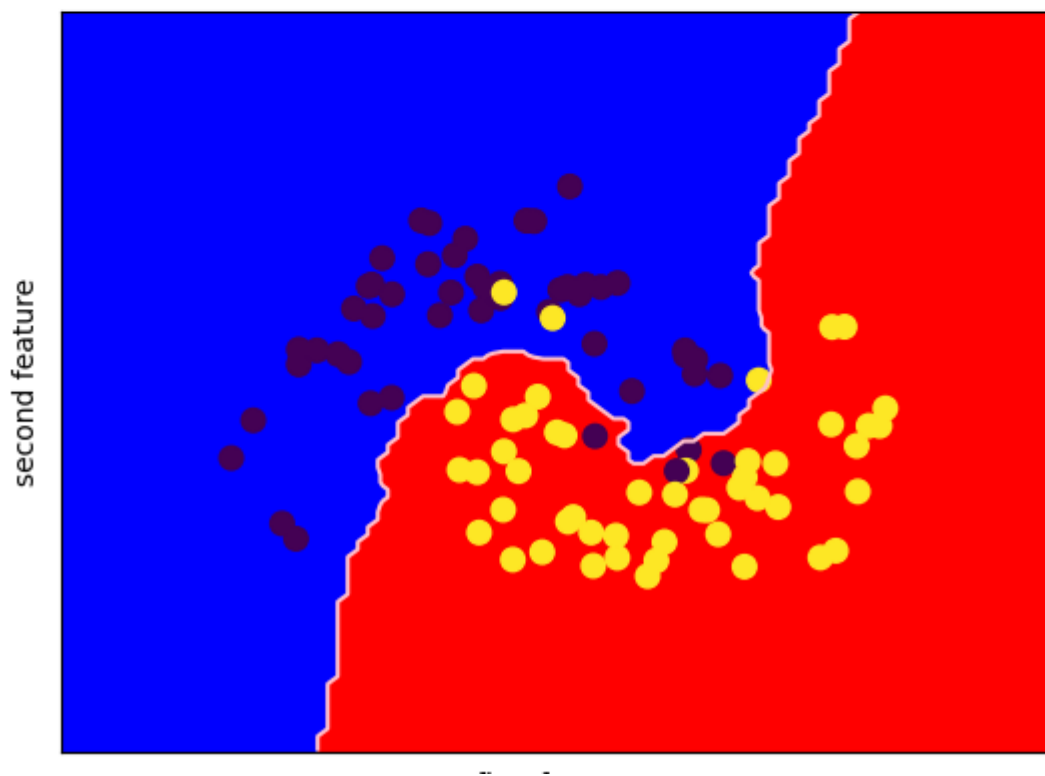


Рисунок 21 – Метод k-ближайших соседей n=5

```

n_neighbors = 9
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
precision    recall  f1-score   support

      0       0.75      0.75      0.75         4
      1       0.83      0.83      0.83         6

 accuracy          0.80         10
macro avg       0.79      0.79      0.79         10
weighted avg    0.80      0.80      0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

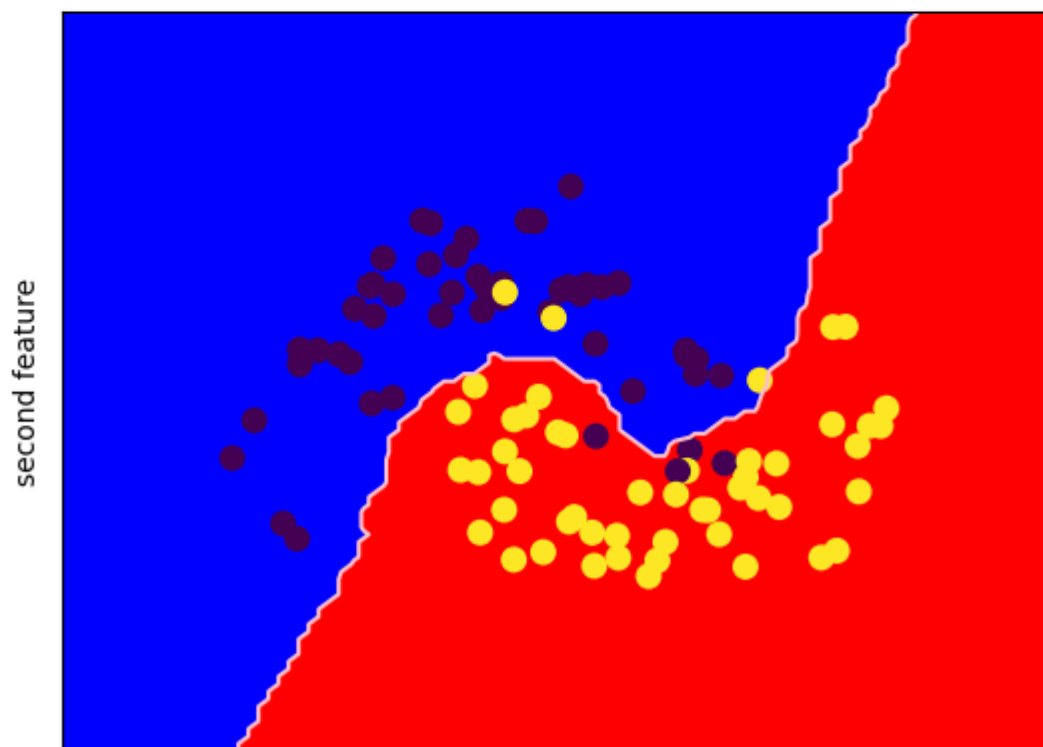


Рисунок 22 – Метод k-ближайших соседей n=9

Предсказанные и истинные значения

[0 1 1 0 0 0 1 1 1 0]

[1 1 1 1 0 0 1 1 0 0]

Матрица ошибок

[[3 1]

[2 4]]

Точность классификации: 0.7

Значения полноты, точности, f1-меры и аккуратности

	precision	recall	f1-score	support
0	0.60	0.75	0.67	4
1	0.80	0.67	0.73	6
accuracy			0.70	10
macro avg	0.70	0.71	0.70	10
weighted avg	0.72	0.70	0.70	10

Значение площади под кривой ошибок (AUC ROC)

0.7083333333333334

Область принятия решений

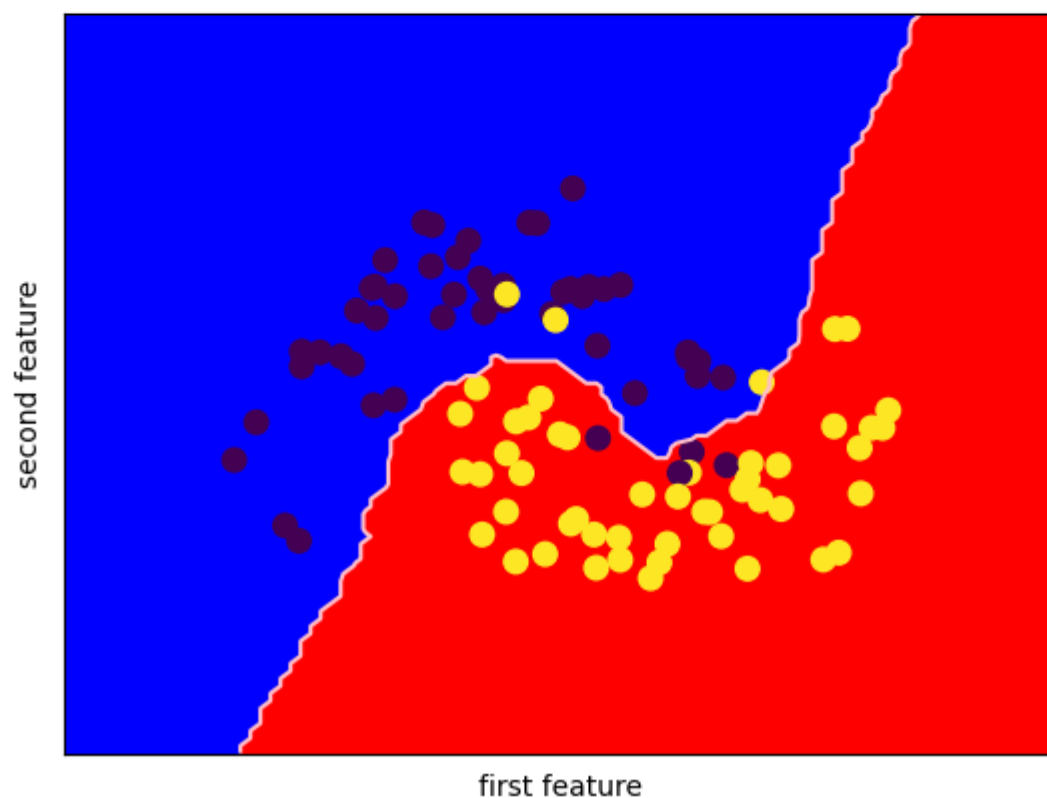


Рисунок 23 – Наивный байесовский метод

```

n_estimators = 5
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0       0.75      0.75      0.75         4
      1       0.83      0.83      0.83         6

 accuracy          0.80         10
 macro avg       0.79      0.79      0.79         10
weighted avg       0.80      0.80      0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

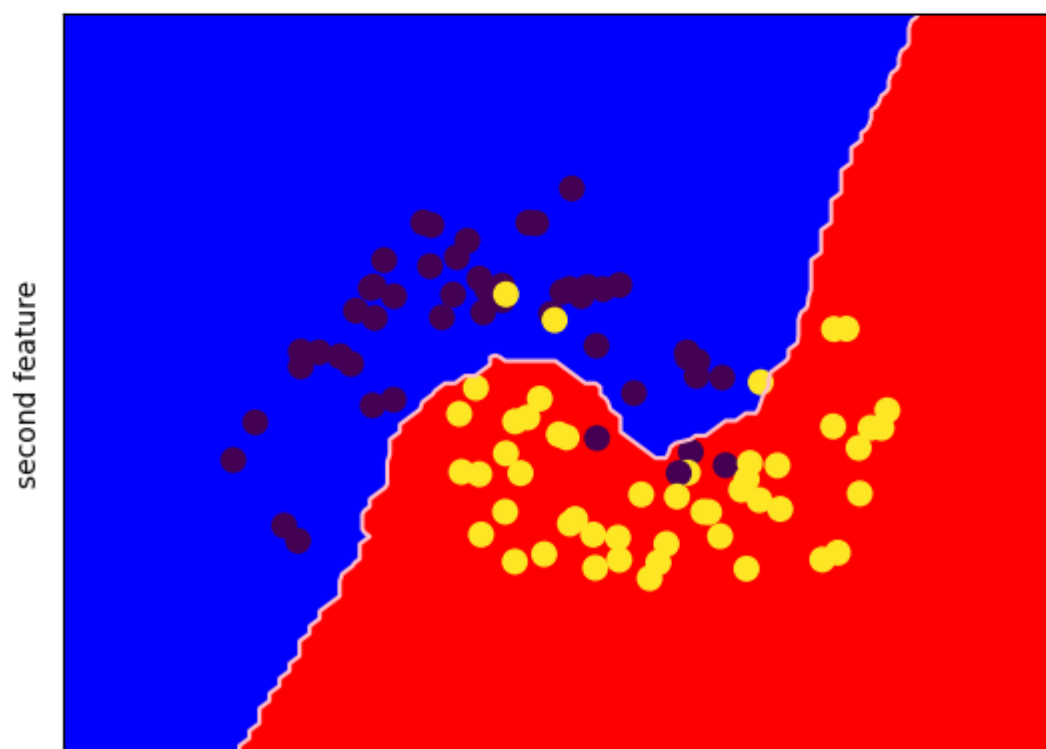


Рисунок 24 – Случайный лес $n = 5$

```

n_estimators = 10
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
precision    recall  f1-score   support

      0       0.75      0.75      0.75        4
      1       0.83      0.83      0.83        6

 accuracy          0.80        10
 macro avg       0.79      0.79      0.79        10
weighted avg       0.80      0.80      0.80        10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

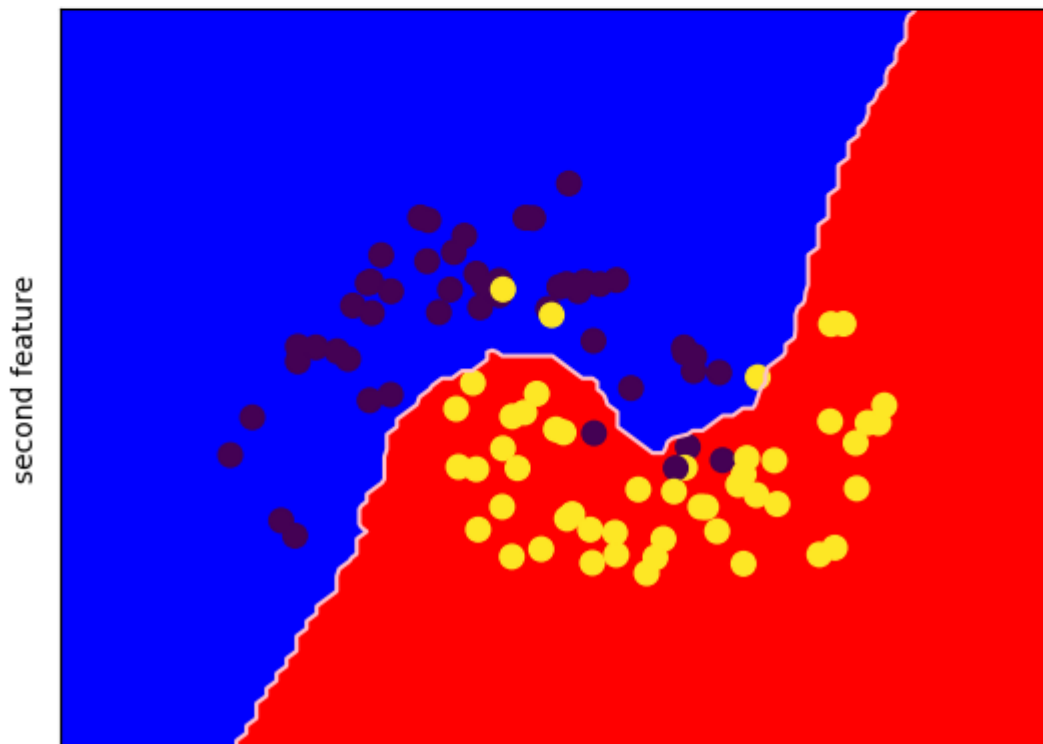


Рисунок 25 – Случайный лес $n = 10$

```

n_estimators = 15
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
precision    recall  f1-score   support

      0       0.75      0.75      0.75         4
      1       0.83      0.83      0.83         6

 accuracy          0.80         10
 macro avg       0.79      0.79      0.79         10
weighted avg       0.80      0.80      0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

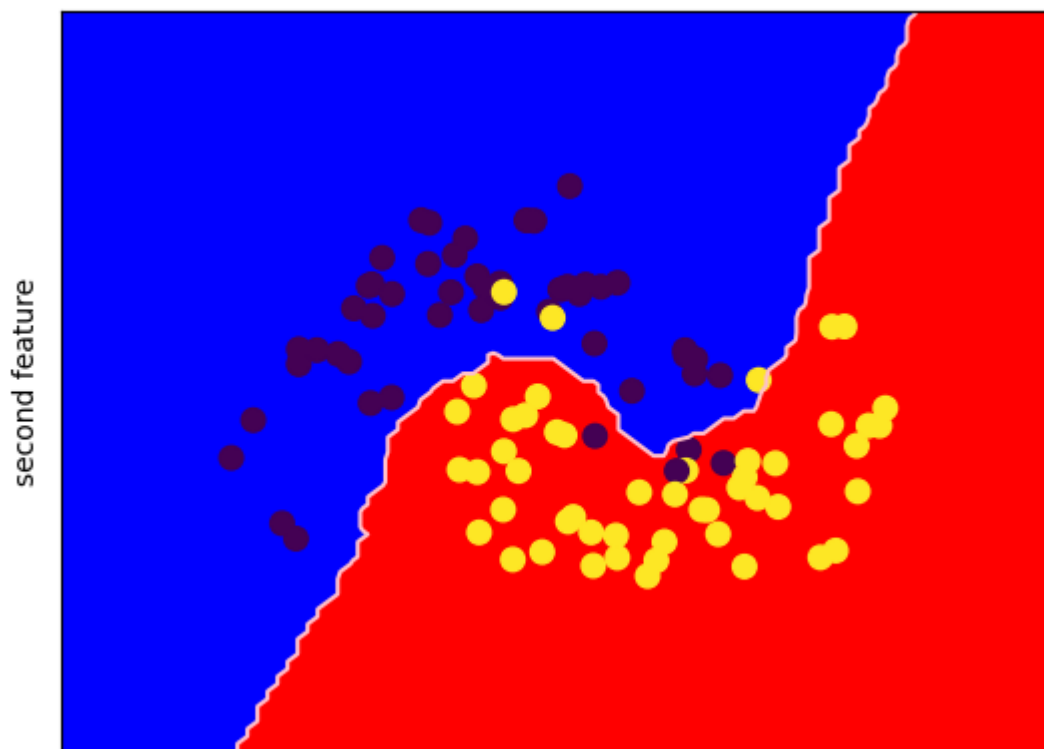


Рисунок 26 – Случайный лес $n = 15$

```

n_estimators = 20
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0       0.75       0.75       0.75         4
      1       0.83       0.83       0.83         6

 accuracy          0.80         10
 macro avg       0.79       0.79       0.79         10
weighted avg       0.80       0.80       0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

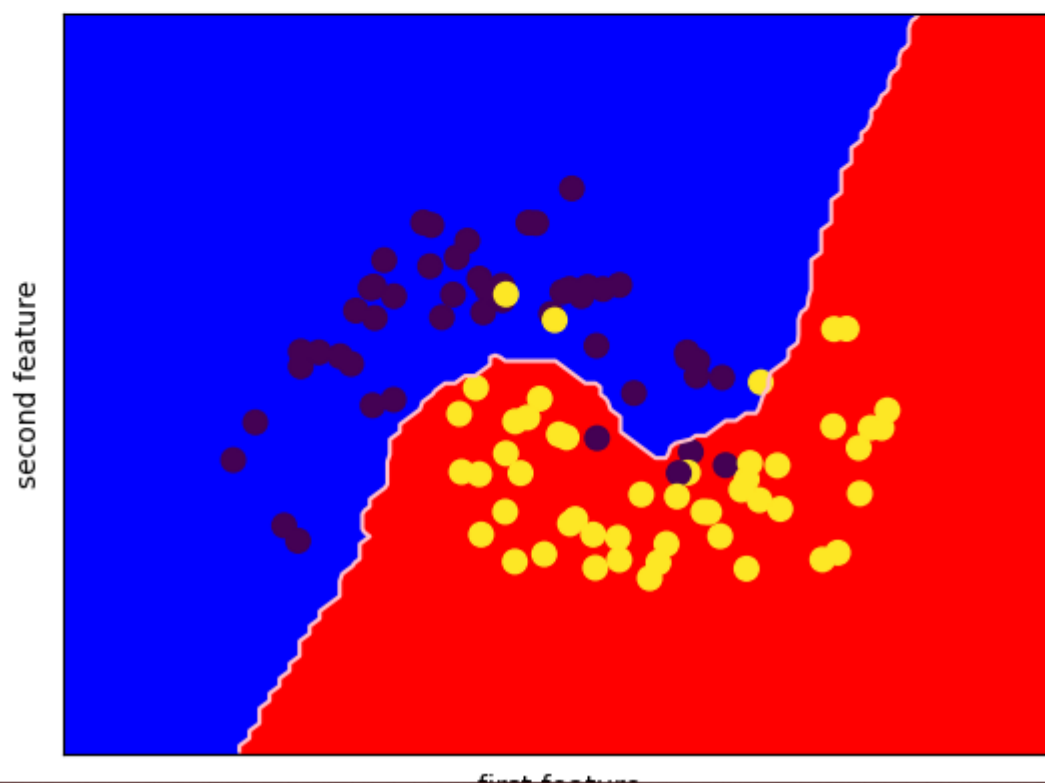


Рисунок 27 – Случайный лес $n = 20$

```

n_estimators = 50
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0]
[1 1 1 1 0 0 1 1 0 0]
Матрица ошибок
[[3 1]
 [1 5]]
Точность классификации: 0.8
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0         0.75      0.75      0.75         4
      1         0.83      0.83      0.83         6

 accuracy          0.80         10
 macro avg         0.79      0.79      0.79         10
 weighted avg      0.80      0.80      0.80         10

Значение площади под кривой ошибок (AUC ROC)
0.7916666666666666
Область принятия решений

```

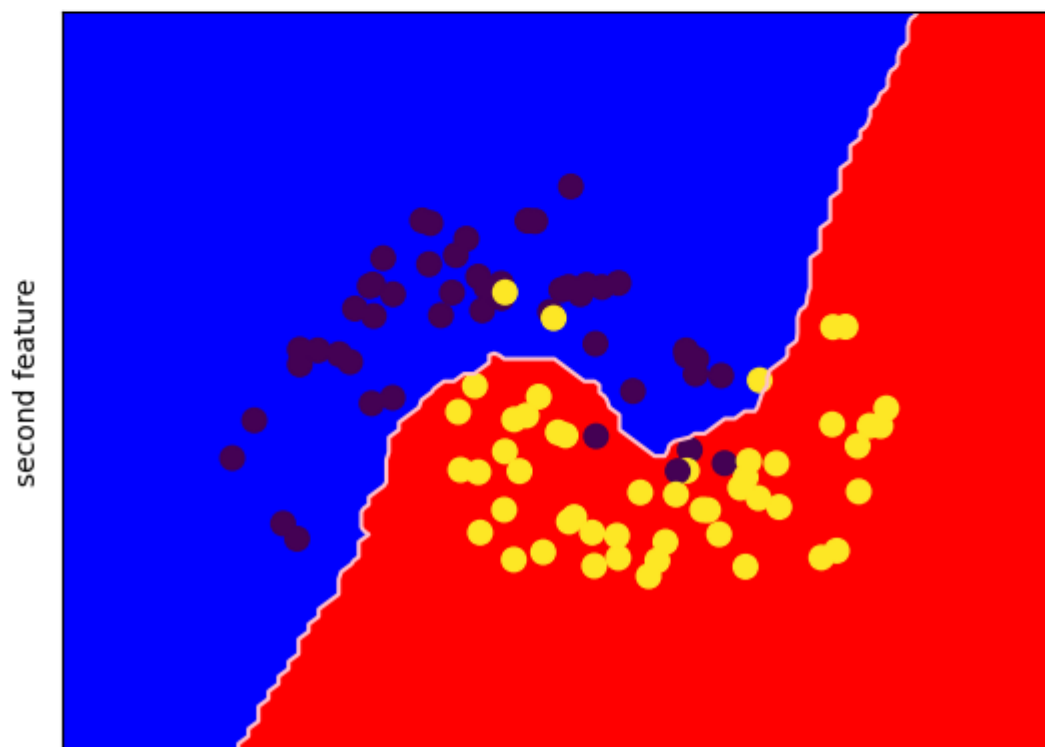


Рисунок 28 – Случайный лес $n = 50$

Таблица 2 – Результаты работы программы

Метод	Истинные и предсказанные метки классов	Матрица ошибок	Значения полноты, точности, f1- меры и аккуратности	Значение площади под кривой ошибок
k-ближайших соседей n=1	$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix}$	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support 0 0.75 0.75 0.75 4 1 0.83 0.83 0.83 6 accuracy 0.80 10 macro avg 0.79 0.79 0.79 10 weighted avg 0.80 0.80 0.80 10	0.7916666666666666
k-ближайших соседей n=3	$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix}$	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support 0 0.75 0.75 0.75 4 1 0.83 0.83 0.83 6 accuracy 0.80 10 macro avg 0.79 0.79 0.79 10 weighted avg 0.80 0.80 0.80 10	0.7916666666666666
k-ближайших соседей n=5	$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix}$	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support 0 0.75 0.75 0.75 4 1 0.83 0.83 0.83 6 accuracy 0.80 10 macro avg 0.79 0.79 0.79 10 weighted avg 0.80 0.80 0.80 10	0.7916666666666666
k-ближайших соседей n=9	$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix}$	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support 0 0.75 0.75 0.75 4 1 0.83 0.83 0.83 6 accuracy 0.80 10 macro avg 0.79 0.79 0.79 10 weighted avg 0.80 0.80 0.80 10	0.7916666666666666

Наивный байесовский метод	<pre> 0 1 1 0 0 0 1 1 1 0 1 1 1 1 0 0 1 1 0 0 </pre>	<pre> [[3 1] [2 4]] </pre>	<div>Значения полноты, точности, f1-меры и аккуратности</div> <table> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> <tr> <td>0</td><td>0.60</td><td>0.75</td><td>0.67</td><td>4</td></tr> <tr> <td>1</td><td>0.80</td><td>0.67</td><td>0.73</td><td>6</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.70</td><td>10</td></tr> <tr> <td>macro avg</td><td>0.70</td><td>0.71</td><td>0.70</td><td>10</td></tr> <tr> <td>weighted avg</td><td>0.72</td><td>0.70</td><td>0.70</td><td>10</td></tr> </table>		precision	recall	f1-score	support	0	0.60	0.75	0.67	4	1	0.80	0.67	0.73	6	accuracy			0.70	10	macro avg	0.70	0.71	0.70	10	weighted avg	0.72	0.70	0.70	10	0.7083333333333334
	precision	recall	f1-score	support																														
0	0.60	0.75	0.67	4																														
1	0.80	0.67	0.73	6																														
accuracy			0.70	10																														
macro avg	0.70	0.71	0.70	10																														
weighted avg	0.72	0.70	0.70	10																														
Случайный лес n=5	<pre> 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 0 1 1 0 0 </pre>	<pre> [[3 1] [1 5]] </pre>	<div>Значения полноты, точности, f1-меры и аккуратности</div> <table> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> <tr> <td>0</td><td>0.75</td><td>0.75</td><td>0.75</td><td>4</td></tr> <tr> <td>1</td><td>0.83</td><td>0.83</td><td>0.83</td><td>6</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.80</td><td>10</td></tr> <tr> <td>macro avg</td><td>0.79</td><td>0.79</td><td>0.79</td><td>10</td></tr> <tr> <td>weighted avg</td><td>0.80</td><td>0.80</td><td>0.80</td><td>10</td></tr> </table>		precision	recall	f1-score	support	0	0.75	0.75	0.75	4	1	0.83	0.83	0.83	6	accuracy			0.80	10	macro avg	0.79	0.79	0.79	10	weighted avg	0.80	0.80	0.80	10	0.7916666666666666
	precision	recall	f1-score	support																														
0	0.75	0.75	0.75	4																														
1	0.83	0.83	0.83	6																														
accuracy			0.80	10																														
macro avg	0.79	0.79	0.79	10																														
weighted avg	0.80	0.80	0.80	10																														
Случайный лес n=10	<pre> 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 0 1 1 0 0 </pre>	<pre> [[3 1] [1 5]] </pre>	<div>Значения полноты, точности, f1-меры и аккуратности</div> <table> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> <tr> <td>0</td><td>0.75</td><td>0.75</td><td>0.75</td><td>4</td></tr> <tr> <td>1</td><td>0.83</td><td>0.83</td><td>0.83</td><td>6</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.80</td><td>10</td></tr> <tr> <td>macro avg</td><td>0.79</td><td>0.79</td><td>0.79</td><td>10</td></tr> <tr> <td>weighted avg</td><td>0.80</td><td>0.80</td><td>0.80</td><td>10</td></tr> </table>		precision	recall	f1-score	support	0	0.75	0.75	0.75	4	1	0.83	0.83	0.83	6	accuracy			0.80	10	macro avg	0.79	0.79	0.79	10	weighted avg	0.80	0.80	0.80	10	0.7916666666666666
	precision	recall	f1-score	support																														
0	0.75	0.75	0.75	4																														
1	0.83	0.83	0.83	6																														
accuracy			0.80	10																														
macro avg	0.79	0.79	0.79	10																														
weighted avg	0.80	0.80	0.80	10																														
Случайный лес n=15	<pre> 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 0 1 1 0 0 </pre>	<pre> [[3 1] [1 5]] </pre>	<div>Значения полноты, точности, f1-меры и аккуратности</div> <table> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> <tr> <td>0</td><td>0.75</td><td>0.75</td><td>0.75</td><td>4</td></tr> <tr> <td>1</td><td>0.83</td><td>0.83</td><td>0.83</td><td>6</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.80</td><td>10</td></tr> <tr> <td>macro avg</td><td>0.79</td><td>0.79</td><td>0.79</td><td>10</td></tr> <tr> <td>weighted avg</td><td>0.80</td><td>0.80</td><td>0.80</td><td>10</td></tr> </table>		precision	recall	f1-score	support	0	0.75	0.75	0.75	4	1	0.83	0.83	0.83	6	accuracy			0.80	10	macro avg	0.79	0.79	0.79	10	weighted avg	0.80	0.80	0.80	10	0.7916666666666666
	precision	recall	f1-score	support																														
0	0.75	0.75	0.75	4																														
1	0.83	0.83	0.83	6																														
accuracy			0.80	10																														
macro avg	0.79	0.79	0.79	10																														
weighted avg	0.80	0.80	0.80	10																														
Случайный лес n=20	<pre> 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 0 1 1 0 0 </pre>	<pre> [[3 1] [1 5]] </pre>	<div>Значения полноты, точности, f1-меры и аккуратности</div> <table> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> <tr> <td>0</td><td>0.75</td><td>0.75</td><td>0.75</td><td>4</td></tr> <tr> <td>1</td><td>0.83</td><td>0.83</td><td>0.83</td><td>6</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>0.80</td><td>10</td></tr> <tr> <td>macro avg</td><td>0.79</td><td>0.79</td><td>0.79</td><td>10</td></tr> <tr> <td>weighted avg</td><td>0.80</td><td>0.80</td><td>0.80</td><td>10</td></tr> </table>		precision	recall	f1-score	support	0	0.75	0.75	0.75	4	1	0.83	0.83	0.83	6	accuracy			0.80	10	macro avg	0.79	0.79	0.79	10	weighted avg	0.80	0.80	0.80	10	0.7916666666666666
	precision	recall	f1-score	support																														
0	0.75	0.75	0.75	4																														
1	0.83	0.83	0.83	6																														
accuracy			0.80	10																														
macro avg	0.79	0.79	0.79	10																														
weighted avg	0.80	0.80	0.80	10																														

		Значения полноты, точности, f1-меры и аккуратности			
		precision	recall	f1-score	support
Случайный лес n=50	0	0.75	0.75	0.75	4
	1	0.83	0.83	0.83	6
	accuracy			0.80	10
	macro avg	0.79	0.79	0.79	10
	weighted avg	0.80	0.80	0.80	10
		0.7916666666666666			
		[[3 1]			
		[1 5]]			

Аккуратность при данном разбиении выборки одинакова во всех случаях, кроме НБК, следовательно все методы, кроме НБК подходят для классификации данных.

Разобьем данные на обучающие (train) и тестовые (test) выборки в пропорции 65% - 35% соответственно

```
: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.35, random_state=1)
```

Рисунок 29 – Разбиение данных на обучающие и тестовые

```
plt.scatter(x_train[:,0], x_train[:,1], c=y_train)
```

```
<matplotlib.collections.PathCollection at 0x1d4eba26a40>
```

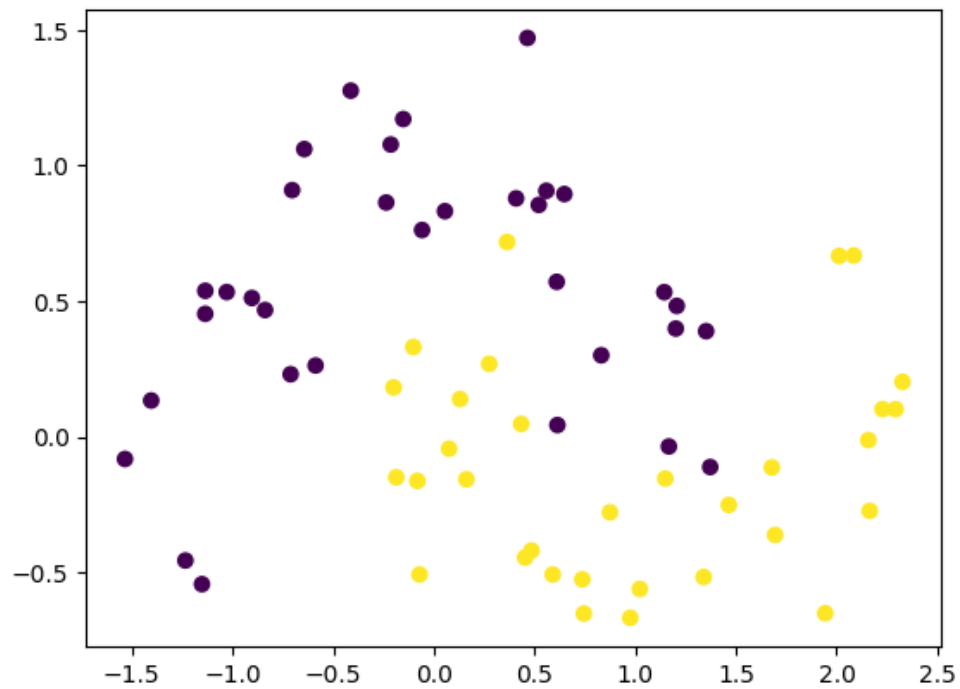


Рисунок 30 – График обучающей выборки

```
plt.scatter(x_test[:,0], x_test[:,1], c=y_test)
```

```
<matplotlib.collections.PathCollection at 0x1d4ebab9240>
```

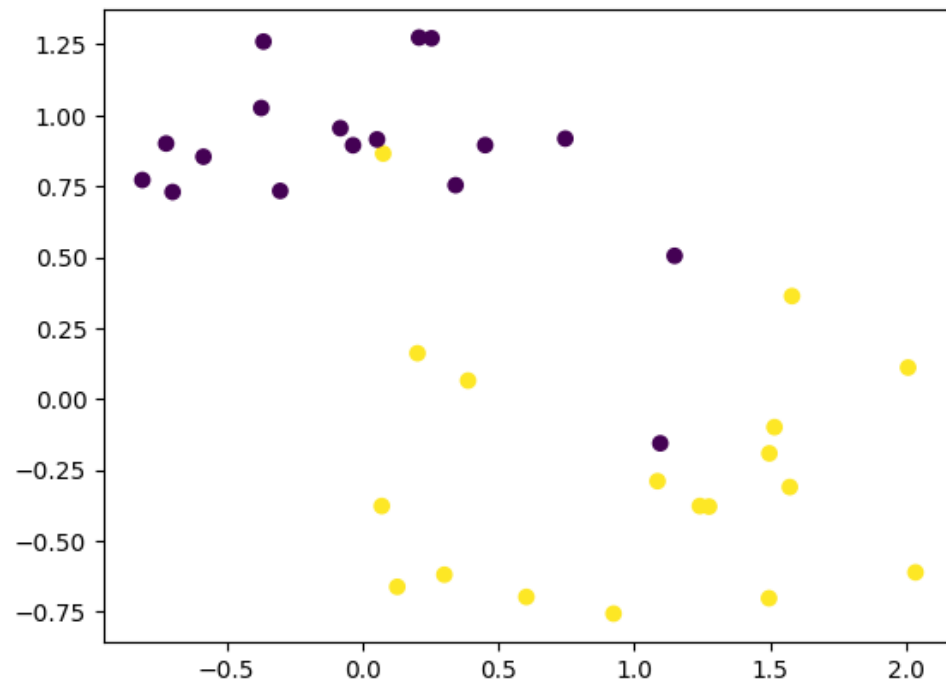


Рисунок 31 – График тестовой выборки

```

n_neighbors = 1
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[15  2]
 [ 3 15]]
Точность классификации: 0.8571428571428571
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.83       0.88       0.86         17
     1       0.88       0.83       0.86         18

 accuracy          0.86         35
 macro avg       0.86       0.86       0.86         35
weighted avg       0.86       0.86       0.86         35

Значение площади под кривой ошибок (AUC ROC)
0.857843137254902
Область принятия решений

```

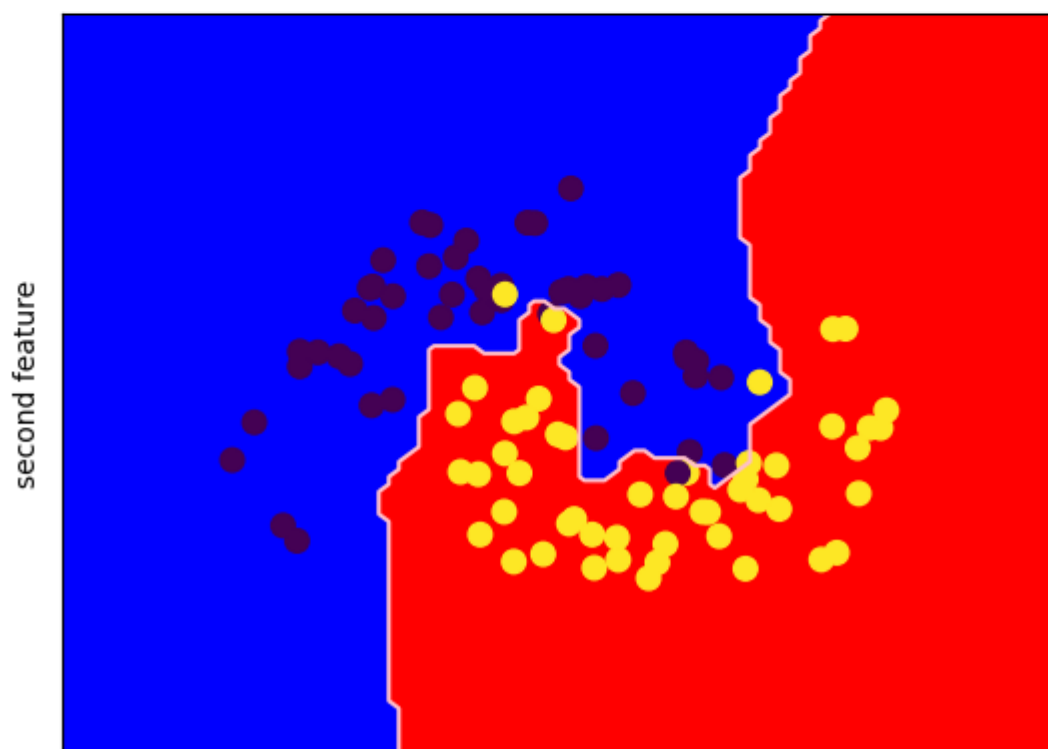


Рисунок 32 – Метод k-ближайших соседей n=1

```

n_neighbors = 3
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 2 16]]
Точность классификации: 0.9142857142857143
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.89       0.94       0.91         17
     1       0.94       0.89       0.91         18

 accuracy          0.91         35
 macro avg       0.92       0.92       0.91         35
weighted avg       0.92       0.91       0.91         35

Значение площади под кривой ошибок (AUC ROC)
0.9150326797385621
Область принятия решений

```

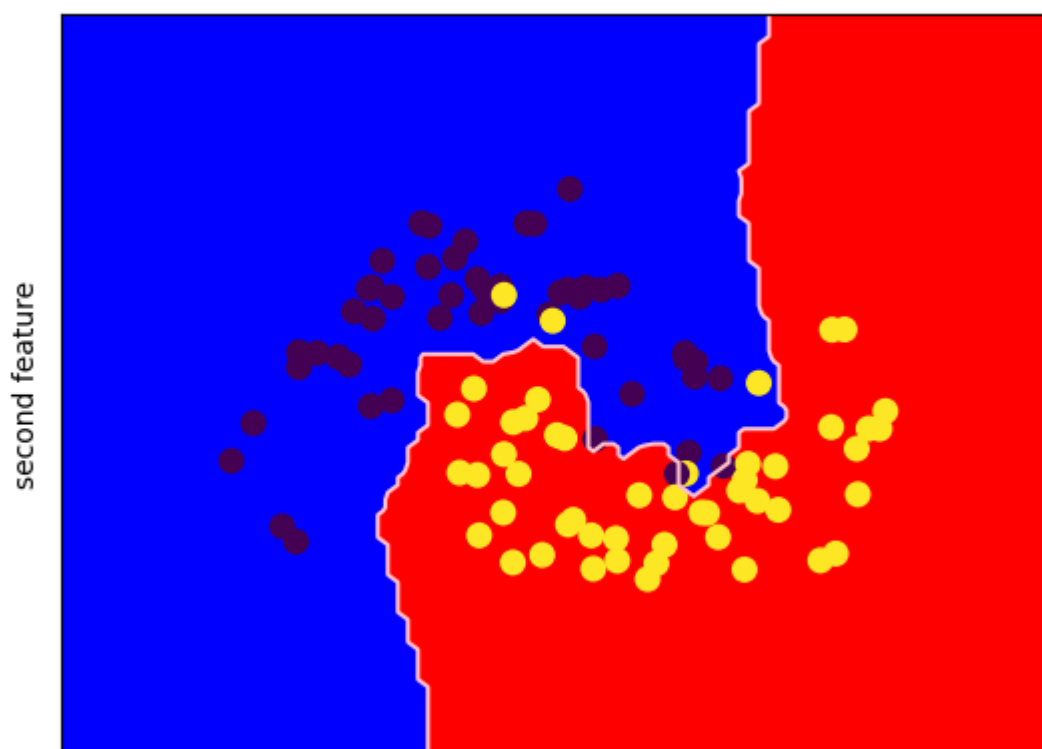


Рисунок 33 – Метод k-ближайших соседей n=3

```

n_neighbors = 5
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 2 16]]
Точность классификации: 0.9142857142857143
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0         0.89      0.94      0.91         17
      1         0.94      0.89      0.91         18

 accuracy          0.91         35
 macro avg         0.92         0.92      0.91         35
weighted avg         0.92         0.91      0.91         35

Значение площади под кривой ошибок (AUC ROC)
0.9150326797385621
Область принятия решений

```

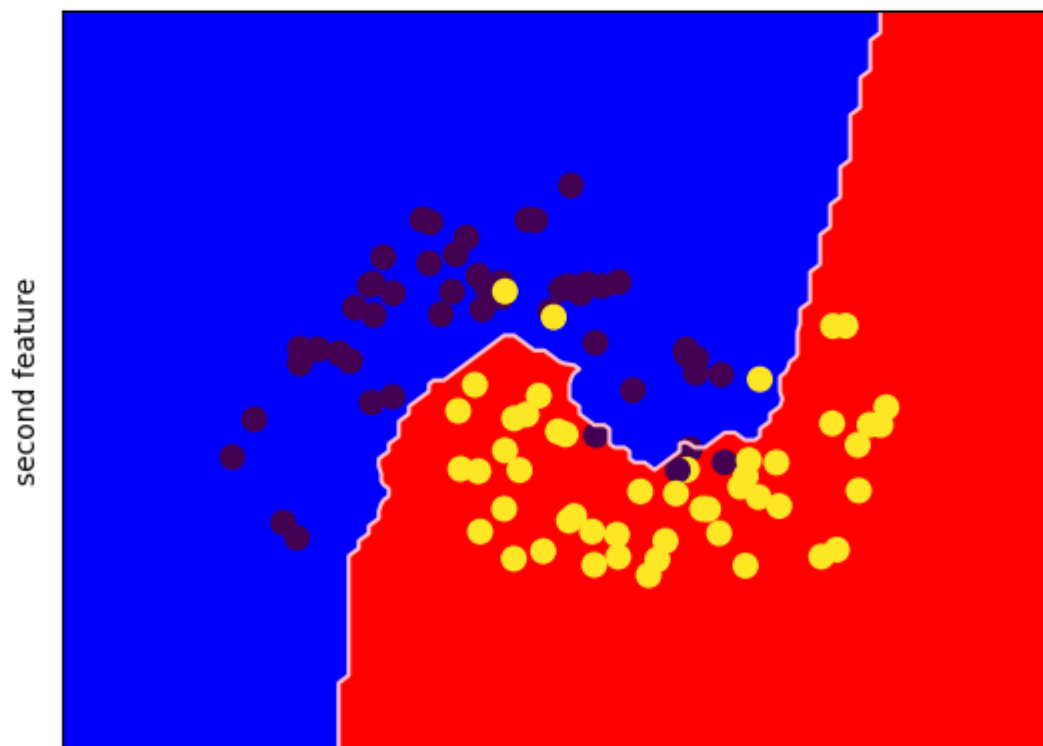


Рисунок 34 – Метод k-ближайших соседей n=5

```

n_neighbors = 9
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 2 16]]
Точность классификации: 0.9142857142857143
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

      0         0.89      0.94      0.91         17
      1         0.94      0.89      0.91         18

 accuracy          0.91         35
 macro avg         0.92         35
weighted avg         0.91         35

Значение площади под кривой ошибок (AUC ROC)
0.9150326797385621
Область принятия решений

```

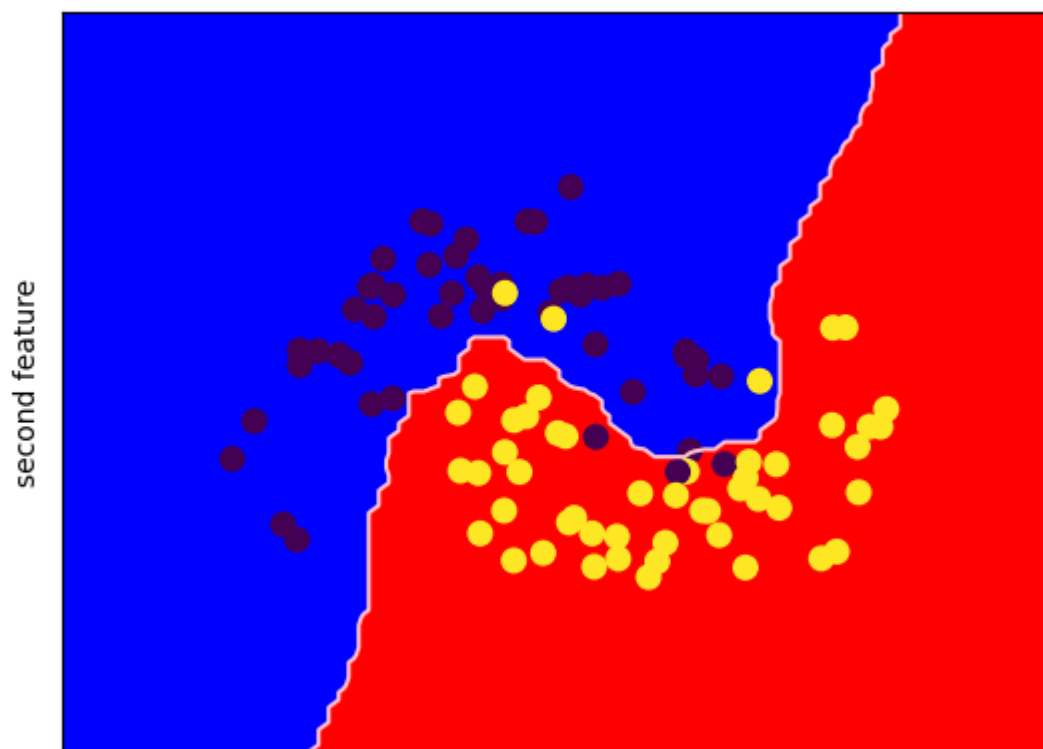


Рисунок 35 – Метод k-ближайших соседей n=9

Предсказанные и истинные значения
 [0 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
 [1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1]
 Матрица ошибок
 [[16 1]
 [2 16]]
 Точность классификации: 0.9142857142857143
 Значения полноты, точности, f1-меры и аккуратности

	precision	recall	f1-score	support
0	0.89	0.94	0.91	17
1	0.94	0.89	0.91	18
accuracy			0.91	35
macro avg	0.92	0.92	0.91	35
weighted avg	0.92	0.91	0.91	35

Значение площади под кривой ошибок (AUC ROC)
 0.9150326797385621
 Область принятия решений

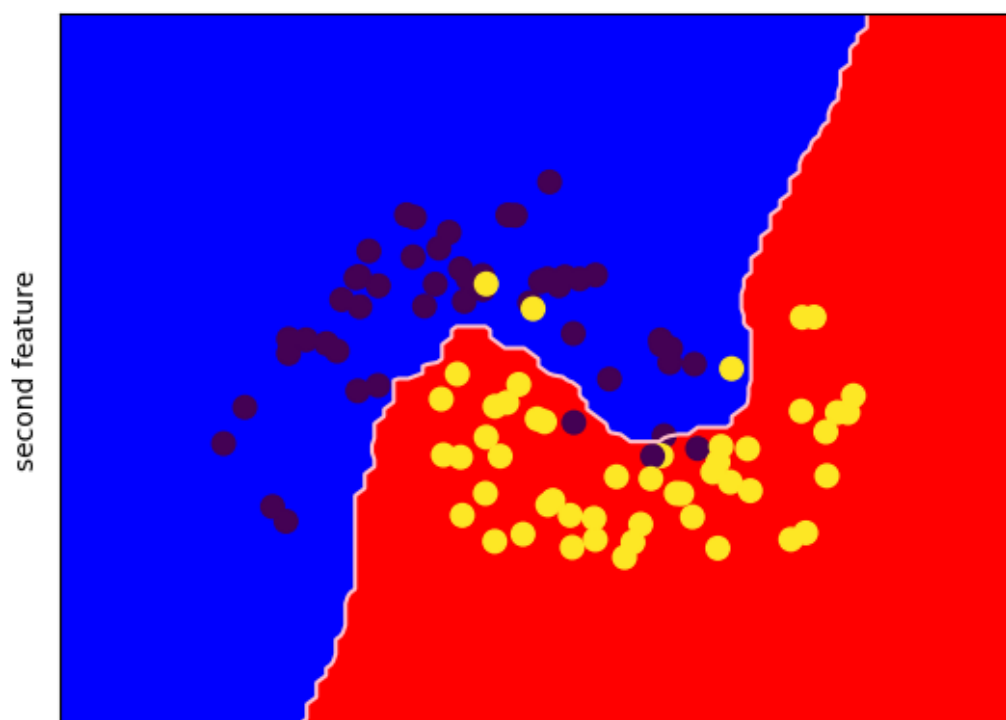


Рисунок 36 – Наивный байесовский классификатор

```

n_estimators = 5
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 3 15]]
Точность классификации: 0.8857142857142857
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.84       0.94       0.89         17
     1       0.94       0.83       0.88         18

 accuracy          0.89         35
 macro avg       0.89       0.89       0.89         35
weighted avg       0.89       0.89       0.89         35

Значение площади под кривой ошибок (AUC ROC)
0.8872549019607844
Область принятия решений

```

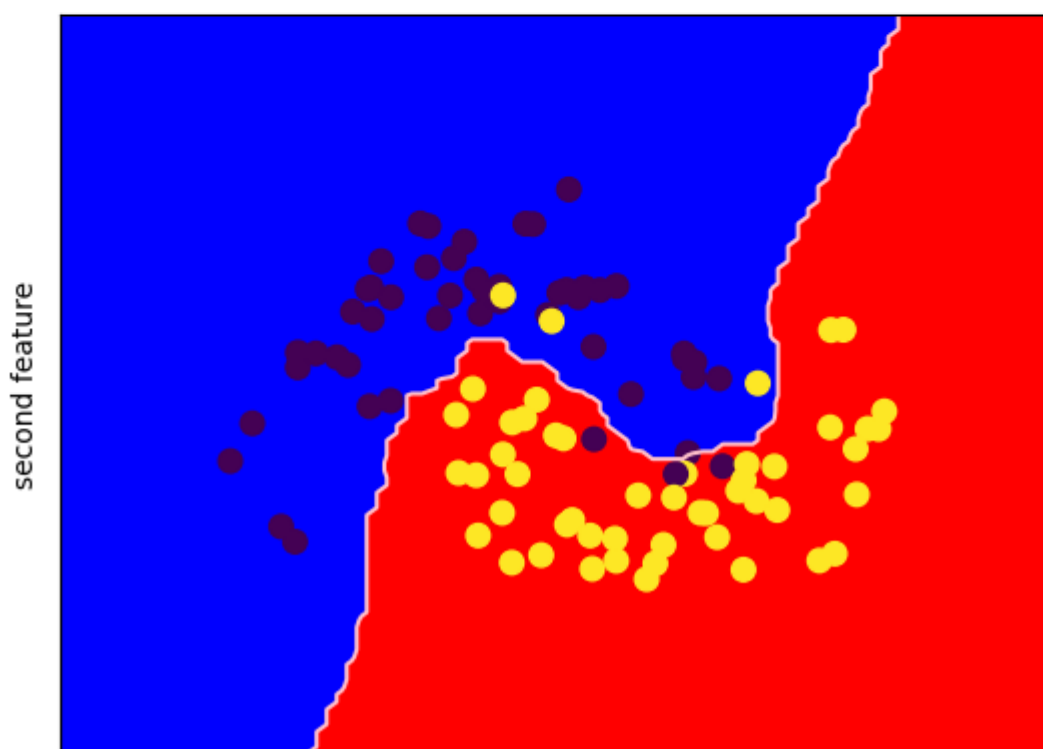


Рисунок 37 – Случайный лес n=5

```

n_estimators = 10
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 3 15]]
Точность классификации: 0.8857142857142857
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.84       0.94       0.89         17
     1       0.94       0.83       0.88         18

 accuracy          0.89         35
 macro avg       0.89       0.89       0.89         35
weighted avg       0.89       0.89       0.89         35

Значение площади под кривой ошибок (AUC ROC)
0.8872549019607844
Область принятия решений

```

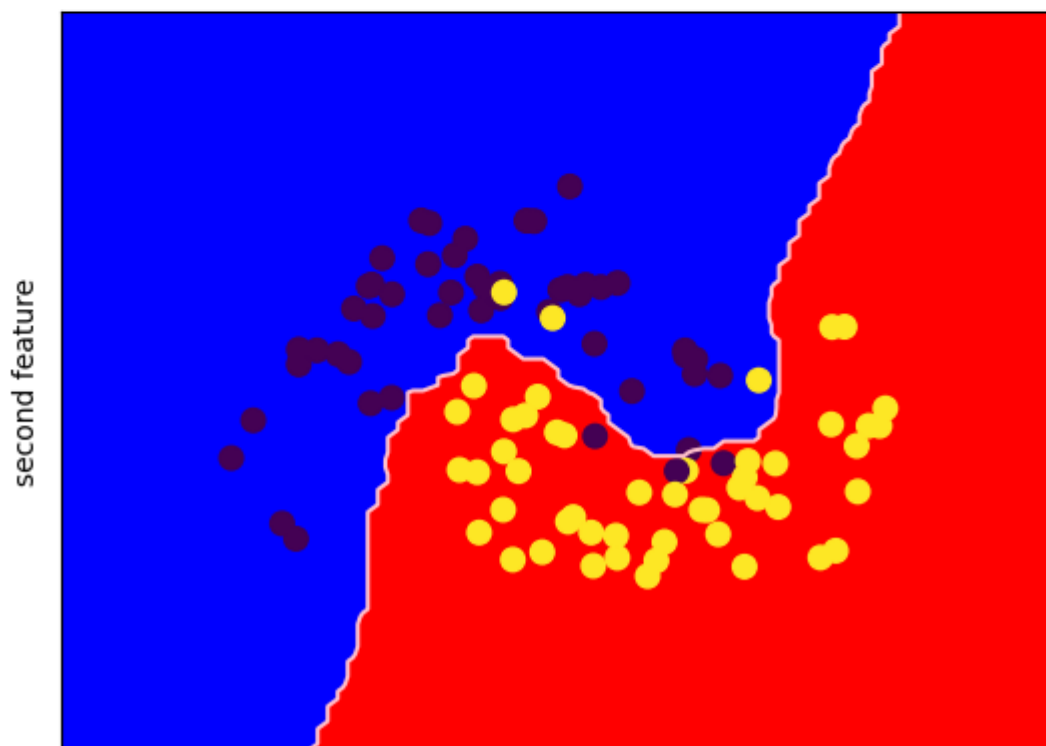


Рисунок 38 – Случайный лес n=10

```

n_estimators = 15
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 3 15]]
Точность классификации: 0.8857142857142857
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0         0.84      0.94      0.89         17
     1         0.94      0.83      0.88         18

 accuracy          0.89         35
 macro avg         0.89      0.89      0.89         35
weighted avg         0.89      0.89      0.89         35

Значение площади под кривой ошибок (AUC ROC)
0.8872549019607844
Область принятия решений

```

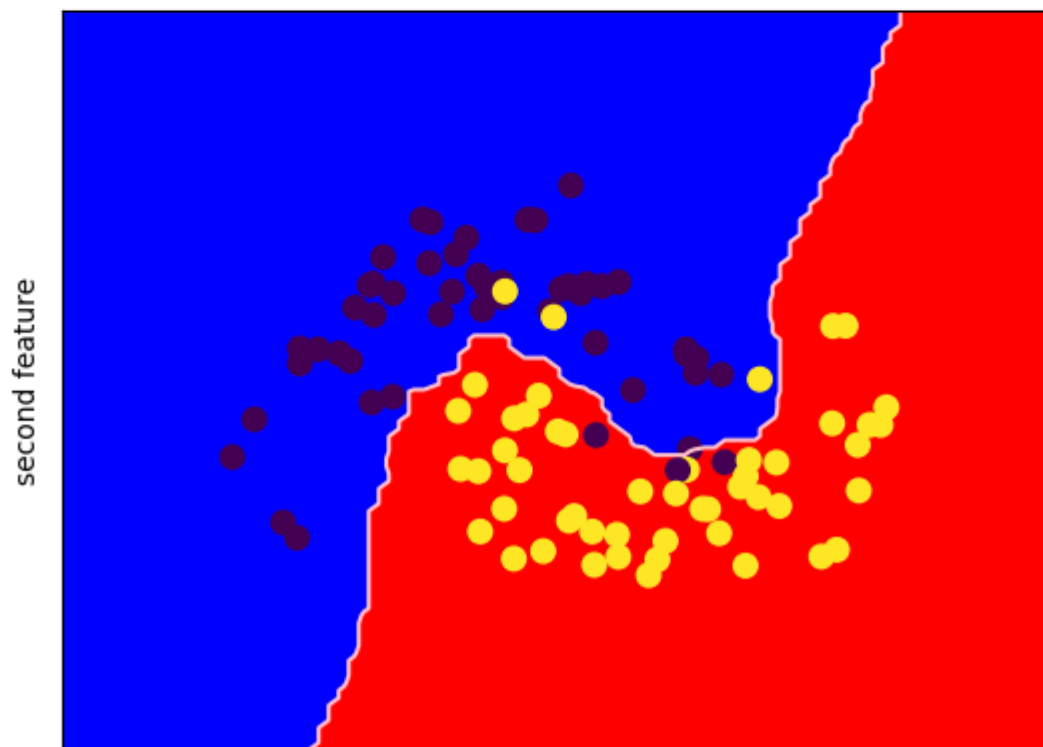


Рисунок 39 – Случайный лес n=15

```

n_estimators = 20
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 1 0 1 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 2 16]]
Точность классификации: 0.9142857142857143
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.89      0.94      0.91         17
     1       0.94      0.89      0.91         18

 accuracy          0.91         35
 macro avg       0.92      0.92      0.91         35
weighted avg       0.92      0.91      0.91         35

Значение площади под кривой ошибок (AUC ROC)
0.9150326797385621
Область принятия решений

```

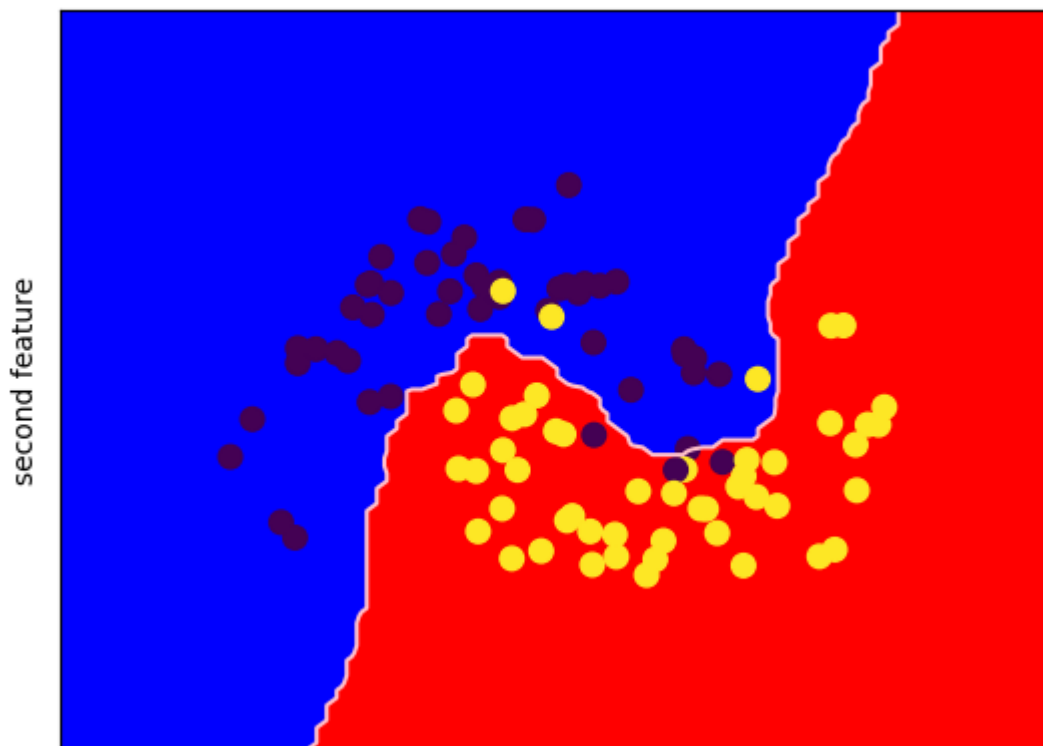


Рисунок 40 – Случайный лес n=20

```

n_estimators = 50
Предсказанные и истинные значения
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1]
Матрица ошибок
[[16  1]
 [ 2 16]]
Точность классификации: 0.9142857142857143
Значения полноты, точности, f1-меры и аккуратности
      precision    recall  f1-score   support

     0       0.89       0.94       0.91         17
     1       0.94       0.89       0.91         18

 accuracy          0.91         35
 macro avg       0.92       0.92       0.91         35
weighted avg       0.92       0.91       0.91         35

Значение площади под кривой ошибок (AUC ROC)
0.9150326797385621
Область принятия решений

```

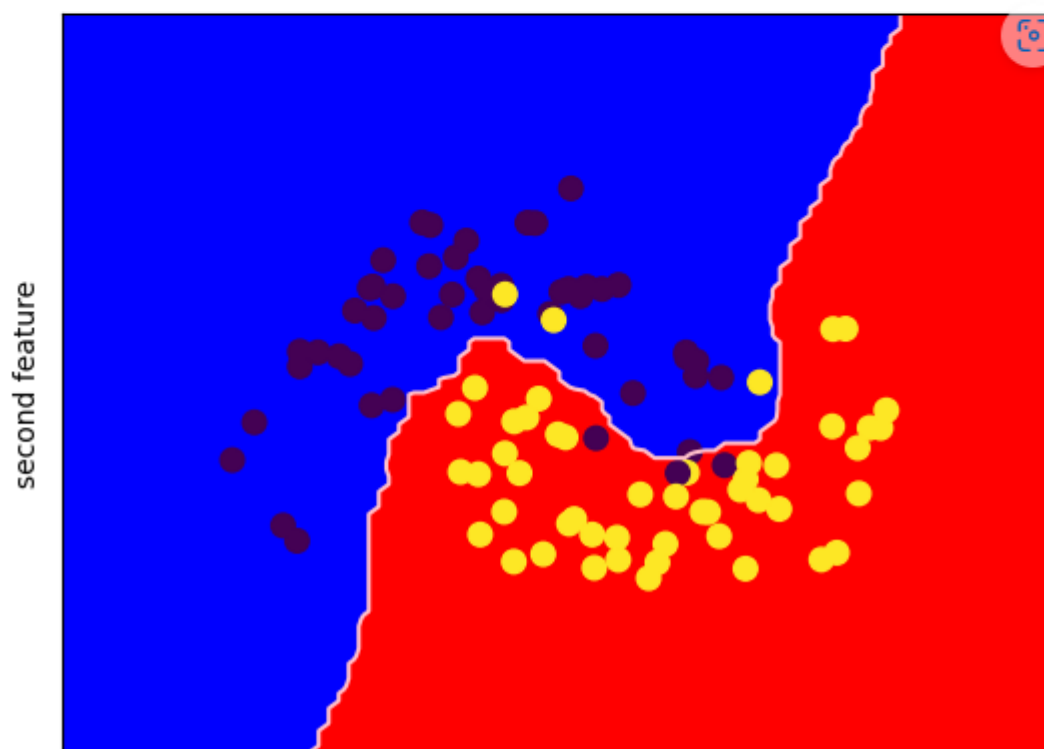


Рисунок 41 – Случайный лес n=50

Таблица 3 – Результаты работы программы

Метод	Истинные и предсказанные метки классов	Матрица ошибок	Значения полноты, точности, f1- меры и аккуратности	Значение площади под кривой ошибок
k-ближайших соседей n=1	[1 1 1 0 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0] [1 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]	[[15 2] [3 15]]	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support	0.857843137254902
			0 0.83 0.88 0.86 17 1 0.88 0.83 0.86 18	
			accuracy 0.86 35 macro avg 0.86 0.86 0.86 35 weighted avg 0.86 0.86 0.86 35	
k-ближайших соседей n=3	[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0] [1 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]	[[16 1] [2 16]]	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support	0.9150326797385621
			0 0.89 0.94 0.91 17 1 0.94 0.89 0.91 18	
			accuracy 0.91 35 macro avg 0.92 0.92 0.91 35 weighted avg 0.92 0.91 0.91 35	
k-ближайших соседей n=5	[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0] [1 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]	[[16 1] [2 16]]	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support	0.9150326797385621
			0 0.89 0.94 0.91 17 1 0.94 0.89 0.91 18	
			accuracy 0.91 35 macro avg 0.92 0.92 0.91 35 weighted avg 0.92 0.91 0.91 35	
k-ближайших соседей n=9	[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0] [1 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]	[[16 1] [2 16]]	Значения полноты, точности, f1-меры и аккуратности precision recall f1-score support	0.9150326797385621
			0 0.89 0.94 0.91 17 1 0.94 0.89 0.91 18	
			accuracy 0.91 35 macro avg 0.92 0.92 0.91 35 weighted avg 0.92 0.91 0.91 35	

Наивный
байесовский
метод

```
[0 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
```

```
[[16 1]
 [ 2 16]]
```

Значения полноты, точности, f1-меры и аккуратности
precision recall f1-score support

```
0      0.89      0.94      0.91      17
1      0.94      0.89      0.91      18
```

0.9150326797385621

```
accuracy      0.91      35
macro avg      0.92      0.92      0.91      35
weighted avg      0.92      0.91      0.91      35
```

Случайный лес
n=5

```
[1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1]
```

```
[[16 1]
 [ 3 15]]
```

Значения полноты, точности, f1-меры и аккуратности
precision recall f1-score support

```
0      0.84      0.94      0.89      17
1      0.94      0.83      0.88      18
```

0.8872549019607844

```
accuracy      0.89      35
macro avg      0.89      0.89      0.89      35
weighted avg      0.89      0.89      0.89      35
```

Случайный лес
n=10

```
[1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1]
```

```
[[16 1]
 [ 3 15]]
```

Значения полноты, точности, f1-меры и аккуратности
precision recall f1-score support

```
0      0.84      0.94      0.89      17
1      0.94      0.83      0.88      18
```

0.8872549019607844

```
accuracy      0.89      35
macro avg      0.89      0.89      0.89      35
weighted avg      0.89      0.89      0.89      35
```

Случайный лес
n=15

```
[1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1]
```

```
[[16 1]
 [ 3 15]]
```

Значения полноты, точности, f1-меры и аккуратности
precision recall f1-score support

```
0      0.84      0.94      0.89      17
1      0.94      0.83      0.88      18
```

0.8872549019607844

```
accuracy      0.89      35
macro avg      0.89      0.89      0.89      35
weighted avg      0.89      0.89      0.89      35
```

Случайный лес
n=20

```
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
```

```
[[16 1]
 [ 2 16]]
```

Значения полноты, точности, f1-меры и аккуратности
precision recall f1-score support

```
0      0.89      0.94      0.91      17
1      0.94      0.89      0.91      18
```

0.9150326797385621

```
accuracy      0.91      35
macro avg      0.92      0.92      0.91      35
weighted avg      0.92      0.91      0.91      35
```



```
[1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0]
[1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1]
```

```
[[16  1]
 [ 2 16]]
```

Значения полноты, точности, f1-меры и аккуратности				
	precision	recall	f1-score	support
0	0.89	0.94	0.91	17
1	0.94	0.89	0.91	18
accuracy			0.91	35
macro avg	0.92	0.92	0.91	35
weighted avg	0.92	0.91	0.91	35

0.9150326797385621

Случайный лес
n=50

Наиболее подходящие методы для классификации данных k-ближайших соседей $n = 3, 5, 9$, наивный байесовский классификатор, случайный лес $n=20$, случайный лес $n=50$, так как в этих методах наибольшая аккуратность.

Код программы:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
```

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB

def plot_2d_separator(classifier, X, fill=False, line=True, ax=None, eps=None):
    if eps is None:
        eps = 1.0
    x_min, x_max = x[:, 0].min() - eps, x[:, 0].max() + eps
    y_min, y_max = x[:, 1].min() - eps, x[:, 1].max() + eps
    xx = np.linspace(x_min, x_max, 100)
    yy = np.linspace(y_min, y_max, 100)
    x1, x2 = np.meshgrid(xx, yy)
    x_grid = np.c_[x1.ravel(), x2.ravel()]
    try:
        decision_values = classifier.decision_function(X_grid)
        levels = [0]
        fill_levels = [decision_values.min(), 0,
            decision_values.max()]
    except AttributeError:
        decision_values = classifier.predict_proba(x_grid)[:, 1]
        levels = [.5]
        fill_levels = [0, .5, 1]
    if ax is None:
        ax = plt.gca()
    if fill:
        ax.contourf(x1, x2, decision_values.reshape(x1.shape),
            levels=fill_levels, colors=['blue', 'red', 'yellow'])
    if line:
        ax.contour(x1, x2, decision_values.reshape(x1.shape), levels=levels,
            colors="pink")
    ax.set_xlim(x_min, x_max)
    ax.set_ylim(y_min, y_max)

```

```

ax.set_xticks(())
ax.set_yticks(())
x, y = make_moons(noise=0.25, random_state=41)
print("Координаты x y:\n", x[:15])
print("Метки класса: ", y[:15])
plt.scatter(x[:,0], x[:,1], c=y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.35,
random_state=1)
plt.scatter(x_train[:,0], x_train[:,1], c=y_train)
plt.scatter(x_test[:,0], x_test[:,1], c=y_test)
def print_classification_metrics(classifier, x, y, prediction, y_test):
    print("Предсказанные и истинные значения")
    print(prediction)
    print(y_test)

    print("Матрица ошибок")
    print(confusion_matrix(y_test, prediction))
    print("Точность классификации: ", accuracy_score(prediction, y_test))
    print("Значения полноты, точности, f1-меры и аккуратности")
    print(classification_report(y_test, prediction))
    print("Значение площади под кривой ошибок (AUC ROC)")
    print(roc_auc_score(y_test, prediction))
    print("Область принятия решений")
    plt.xlabel("first feature")
    plt.ylabel("second feature")
    plot_2d_separator(knn, x, fill=True)
    plt.scatter(x[:, 0], x[:, 1], c=y, s=70)
    plt.show()
for i in [1, 3, 5, 9]:
    knn = KNeighborsClassifier(n_neighbors=i, metric='euclidean')

```

```
knn.fit(x_train, y_train)
prediction = knn.predict(x_test)
print("n_neighbors = ", i)
print_classification_metrics(knn, x, y, prediction, y_test)

naive = GaussianNB()
naive.fit(x_train, y_train)
predict = naive.predict(x_test)
print_classification_metrics(naive, x, y, predict, y_test)

for i in [5, 10, 15, 20, 50]:
    rand_forest = RandomForestClassifier(n_estimators=i)
    rand_forest.fit(x_train, y_train)
    prediction = rand_forest.predict(x_test)
    print("n_estimators = ", i)
    print_classification_metrics(knn, x, y, prediction, y_test)
```

Вывод

В ходе выполнения данной лабораторной работы мы получили базовые навыки работы с языком python и набором функций для анализа и обработки данных. Получили практические навыки решения задачи бинарной классификации данных в среде Jupiter Notebook. Научились загружать данные, обучать классификаторы и проводить классификацию. Научились оценивать точность полученных моделей.