

ЛАБОРАТОРНА РОБОТА № 1

ПОПЕРЕДНЯ ОБРОБКА ТА КОНТРОЛЬОВАНА КЛАСИФІКАЦІЯ ДАНИХ

Мета роботи: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити попередню обробку та класифікацію даних.

Хід роботи

Завдання 1. Попередня обробка даних

```
C:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.10.exe "D:/LabsPolI/AI/Lab1/Task 1/1.py"

Binarized data:
[[1. 0. 1.]
 [0. 1. 0.]
 [1. 0. 0.]
 [1. 0. 0.]]

BEFORE:
Mean = [ 3.775 -1.15 -1.3 ]
Std deviation = [3.12039661 6.36651396 4.0620192 ]

AFTER:
Mean = [1.11022302e-16 0.00000000e+00 2.77555756e-17]
Std deviation = [1. 1. 1.]

Min max scaled data:
[[0.74117647 0.39548023 1.          ]
 [0.          1.          0.          ]
 [0.6         0.5819209  0.87234043]
 [1.          0.          0.17021277]]

l1 normalized data:
[[ 0.45132743 -0.25663717  0.2920354 ]
 [-0.0794702  0.51655629 -0.40397351]
 [ 0.609375   0.0625    0.328125  ]
 [ 0.33640553 -0.4562212 -0.20737327]]

l2 normalized data:
[[ 0.75765788 -0.43082507  0.49024922]
 [-0.12030718  0.78199664 -0.61156148]
 [ 0.87690281  0.08993875  0.47217844]
 [ 0.55734935 -0.75585734 -0.34357152]]

Process finished with exit code 0
```

Рис. 1

L1-нормалізація використовує метод найменших абсолютних відхилень, що забезпечує рівність 1 суми абсолютних значень в кожному ряду. L2-

нормалізація використовує метод найменших квадратів, що забезпечує рівність 1 суми квадратів 4 значень.



```
C:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.10.exe "D:/LabsPoli/AI/Lab1/Task 1/2.py"

Label mapping:
black --> 0
green --> 1
red --> 2
white --> 3
yellow --> 4

Labels = ['green', 'red', 'black']
Encoded values = [1, 2, 0]

Encoded values = [3, 0, 4, 1]
Decoded labels = ['white', 'black', 'yellow', 'green']

Process finished with exit code 0
```

Рис. 2

Завдання 2. Попередня обробка нових даних

```

C:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.10.exe D:/LabsPoli/AI/Lab1/task_2.py

Binarized data:
[[0. 1. 1.]
 [1. 1. 0.]
 [0. 1. 1.]
 [0. 0. 0.]]

BEFORE:
Mean = [-0.3  2.3  0.2]
Std deviation = [3.78219513  3.62973828  5.01547605]

AFTER:
Mean = [-5.55111512e-17  5.55111512e-17  0.00000000e+00]
Std deviation = [1. 1. 1.]

Min max scaled data:
[[0.64356436 0.73737374 1.          ]
 [1.          0.56565657 0.07894737]
 [0.2970297  1.          0.81578947]
 [0.          0.          0.          ]]

l1 normalized data:
[[ 0.11403509  0.34210526  0.54385965]
 [ 0.42982456  0.19298246 -0.37719298]
 [-0.171875   0.5078125   0.3203125  ]
 [-0.37681159 -0.24637681 -0.37681159]]

l2 normalized data:
[[ 0.17475265  0.52425796  0.83343572]
 [ 0.71216718  0.31974853 -0.62496303]
 [-0.2752151  0.81313551  0.51290086]
 [-0.64182859 -0.41965715 -0.64182859]]

Process finished with exit code 0

```

Рис. 3

Завдання 3. Класифікація логістичною регресією або логістичний класифікатор

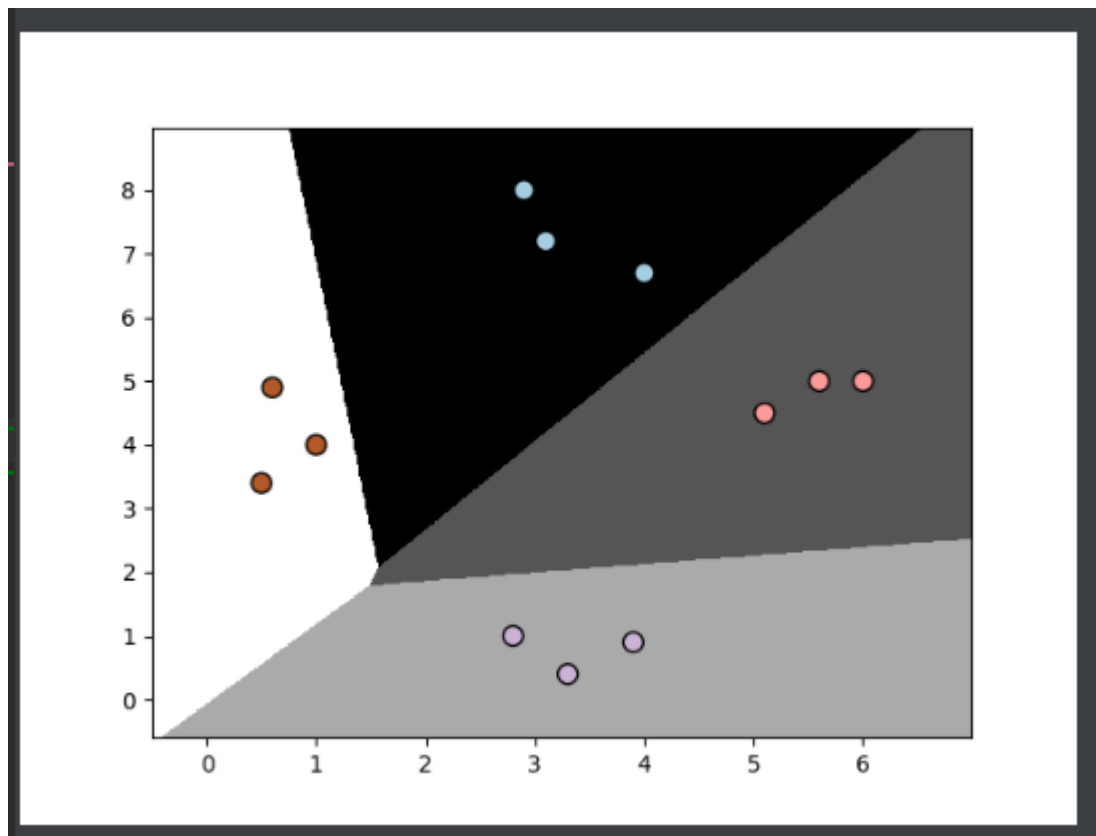


Рис. 4

Завдання 4. Класифікація найвним байєсовським класифікатором

```
LR_1_task_4 x
C:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.10.exe D:/LabsPoli/AI/Lab1/LR_1_task_4.py
Accuracy of Naive Bayes classifier = 99.75 %
Accuracy of the new classifier = 100.0 %
Accuracy: 99.75%
Precision: 99.76%
Recall: 99.75%
F1: 99.75%

Process finished with exit code 0
```

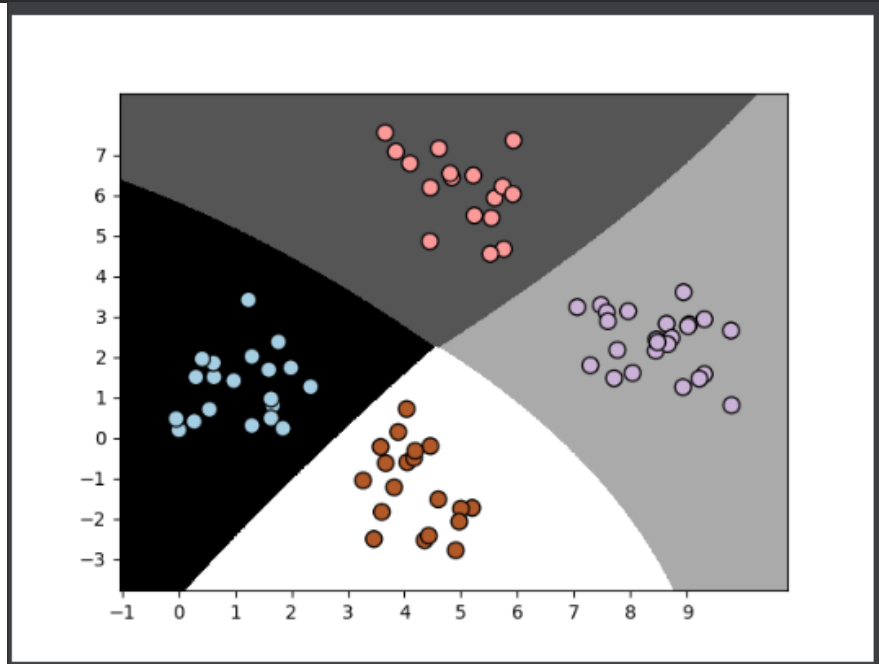


Рис. 5

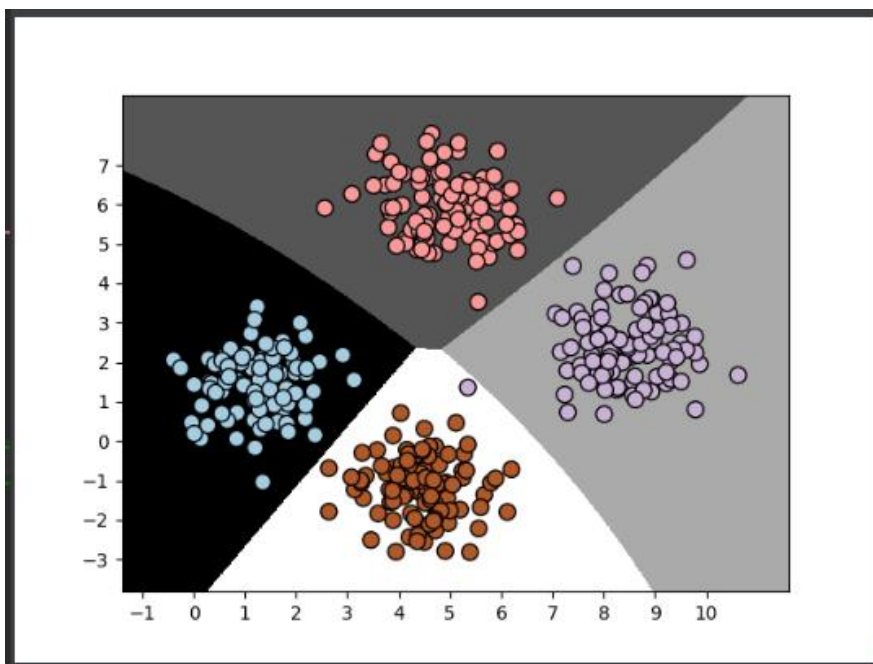


Рис. 6

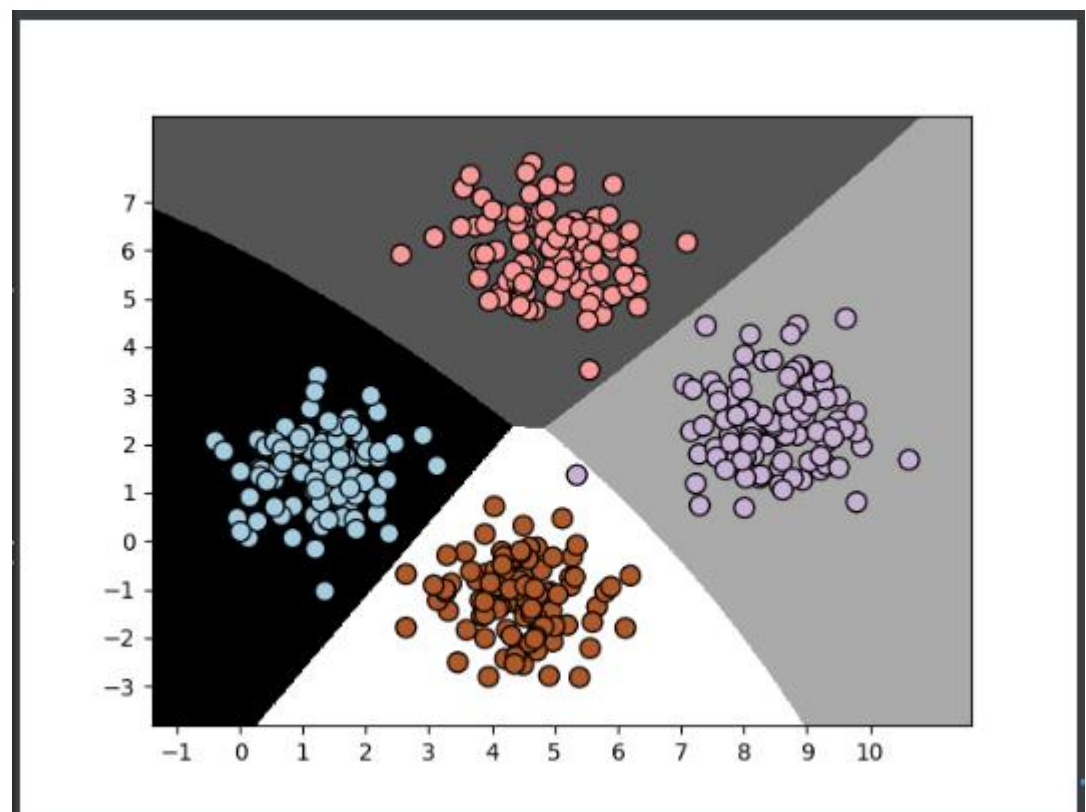
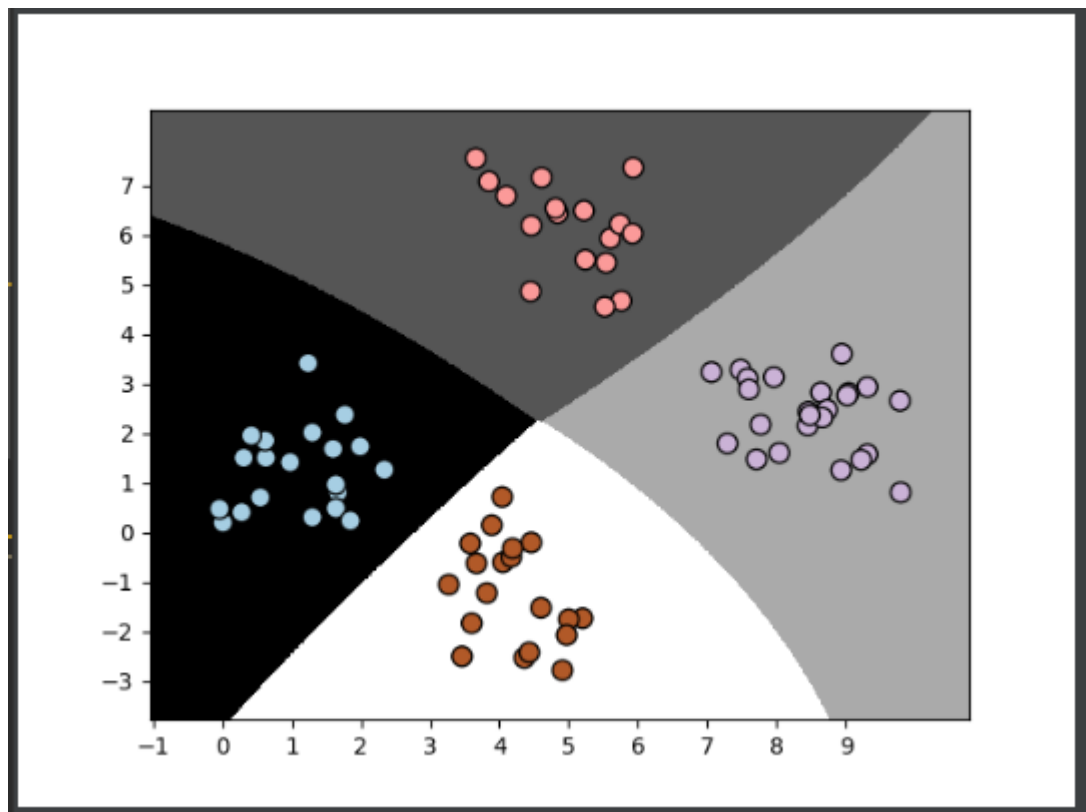


Рис. 7

Завдання 5. Вивчити метрики якості класифікації

```
C:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.10.exe D:/LabsPoli/AI/Lab1/LR_1_task_5.py
TP: 5047
FN: 2832
FP: 2360
TN: 5519
kyiashenko_confusion_matrix: [[5519 2360]
 [2832 5047]]
accuracy_score: 0.6705165630156111
Accuracy RF:0.671
precision_score: 0.681382476036182
recall_score: 0.6405635232897576
Recall RF: 0.641
Precision RF: 0.681
Precision LR: 0.636
f1_score: 0.660342797330891
F1 RF: 0.660
F1 LR: 0.586
scores with threshold = 0.5
Accuracy RF:0.671
Recall RF: 0.641
Precision RF:0.681
F1 RF: 0.660

scores with threshold = 0.25
Accuracy RF:0.502
Recall RF: 1.000
Precision RF:0.501
F1 RF: 0.668

Process finished with exit code 0
```

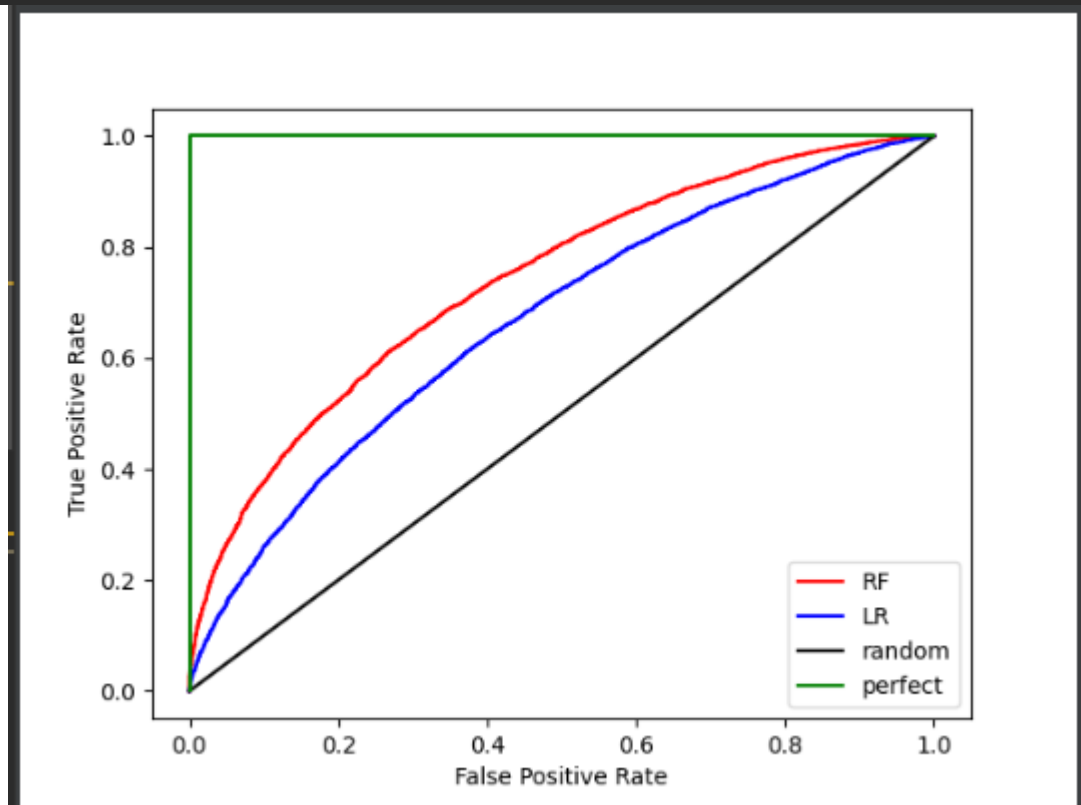


Рис. 8

RF модель показала себе більш точною.

Завдання 6

Розробіть програму класифікації даних в файлі `data_multivar_nb.txt` за допомогою машини опорних векторів (Support Vector Machine - SVM). Розрахуйте показники якості класифікації. Порівняйте їх з показниками наївного байєсівського класифікатора. Зробіть висновки яку модель класифікації краще обрати і чому.


```
C:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.10.exe D:/LabsPoli/AI/Lab1/LR_1_task_6.py
Accuracy of svc classifier = 99.75 %
Accuracy of the new classifier = 100.0 %
Accuracy: 99.75%
Precision: 99.76%
Recall: 99.75%
F1: 99.75%

Process finished with exit code 0
```

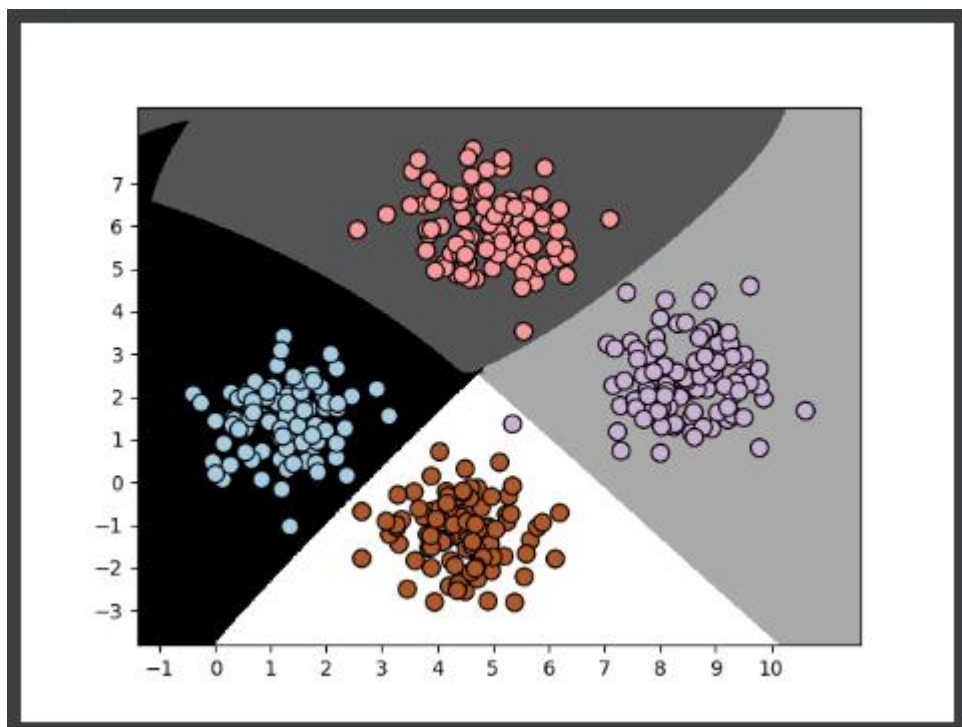
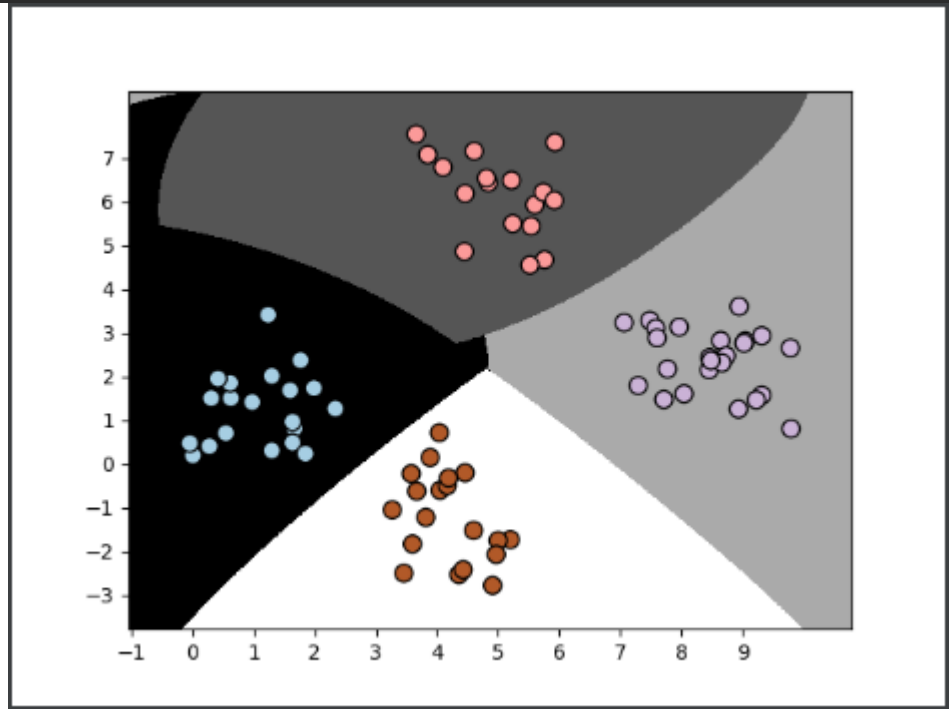


Рис. 9

Метод наївного байєсівського класифікатора спрацював точніше і швидше.

Посилання на Git: <https://github.com/Grum74/AI>

Висновок

Я, використовуючи спеціалізовані бібліотеки та мову програмування Python дослідив попередню обробку та класифікацію даних.