

ЛАБОРАТОРНА РОБОТА № 1

ПОПЕРЕДНЯ ОБРОБКА ТА КОНТРОЛЬОВАНА КЛАСИФІКАЦІЯ ДАНИХ

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

Хід роботи:

Завдання 1: Попередня обробка даних.

Результат:

```

task.py > ...
1  import numpy as np
2  from sklearn import preprocessing
3
4  input_data = np.array([[5.1, -2.9, 3.3],
5                        [-1.2, 7.8, -6.1],
6                        [3.9, 0.4, 2.1],
7                        [7.3, -9.9, -4.5]])
8
9  data_binarized = preprocessing.Binarizer(threshold=2.1).transform(input_data)
10 print("\n Binarized data:\n", data_binarized)
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS D:\DIMA_Навчання\Системи Штучного Інтелекту\lr1> & 'C:\Python310\python.exe' 'c:\Users\kcher' '59053' '--' 'd:\DIMA_Навчання\Системи Штучного Інтелекту\lr1\task.py'

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

Рис. 1.1. Бінаризація

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР01		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Фещенко Д.М.			Звіт з лабораторної роботи 1	Літ.	Арк.
Перевір.		Пулеко І.В.					1
Керівник						ФІКТ Гр. ПІ-59(2)	
Н. контр.							
Затверд.							

```

task.py > ...
1  import numpy as np
2  from sklearn import preprocessing
3
4  input_data = np.array([[5.1, -2.9, 3.3],
5                        [-1.2, 7.8, -6.1],
6                        [3.9, 0.4, 2.1],
7                        [7.3, -9.9, -4.5]])
8
9  data_binarized = preprocessing.Binarizer(threshold=2.1).transform(input_data)
10 print("\n Binarized data:\n", data_binarized)
11
12 print("\nBEFORE: ")
13 print("Mean =", input_data.mean(axis=0))
14 print("Std deviation =", input_data.std(axis=0))
15
16 data_scaled = preprocessing.scale(input_data)
17 print("\nAFTER: ")
18 print("Mean =", data_scaled.mean(axis=0))
19 print("Std deviation =", data_scaled.std(axis=0))
20
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

python.python-2022.6.2\pythonFiles\lib\python\debugpy\launcher '59111' '--' 'd:\ДИМА_Навчання\Системи Штучного Інтелекту\lr1\task.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.]
PS D:\ДИМА_Навчання\Системи Штучного Інтелекту\lr1>

```

Рис. 1.2. Виключення середнього

```

21  data_scaler_minmax = preprocessing.MinMaxScaler(feature_range=(0, 1))
22  data_scaled_minmax = data_scaler_minmax.fit_transform(input_data)
23  print("\nMin max scaled data:\n", data_scaled_minmax)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Min max scaled data:
[[0.74117647 0.39548023 1.          ]
 [0.          1.          0.          ]
 [0.6         0.5819209   0.87234043]
 [1.          0.          0.17021277]]
PS D:\ДИМА_Навчання\Системи Штучного Інтелекту\lr1>

```

Рис. 1.3. Масштабування

```

25 data_normalized_l1 = preprocessing.normalize(input_data, norm='l1')
26 data_normalized_l2 = preprocessing.normalize(input_data, norm='l2')
27 print("\nl1 normalized data:\n", data_normalized_l1)
28 print("\nl2 normalized data:\n", data_normalized_l2)

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```

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]]
PS D:\DIMA_Навчання\Системи Штучного Інтелекту\lr1> 

```

Рис. 1.4. Нормалізація даних

L1-нормалізація використовує метод найменших абсолютних відхилень (Least Absolute Deviations), що забезпечує рівність 1 суми абсолютних значень в кожному ряду.

L2-нормалізація використовує метод найменших квадратів, що забезпечує рівність 1 суми квадратів 4 значень.

Звідси можна зробити висновок, що L1-нормалізація є надійнішою за L2-нормалізацію.

```

LR_1_task_1.py > ...
1  import numpy as np
2  from sklearn import preprocessing
3
4  Input_labels = ['red', 'black', 'red', 'green', 'black']
5
6  encoder = preprocessing.LabelEncoder()
7  encoder.fit(Input_labels)
8
9  print("\nLabel mapping:")
10 for i, item in enumerate(encoder.classes_):
11     print(item, '-->', i)
12
13 test_labels = ['green', 'red', 'black']
14 encoded_values = encoder.transform(test_labels)
15 print("\nLabels =", test_labels)
16 print("Encoded values =", list(encoded_values))
17
18 encoded_values = [3, 0, 4, 1]
19 decoded_list = encoder.inverse_transform(encoded_values)
20 print("\nEncoded values =", encoded_values)
21 print("Decoded labels =", list(decoded_list))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS D:\DIMA_Навчання\Системи Штучного Інтелекту\lr1> d.; cd 'd:\DIMA_Навчання\Системи Штучного Інтелекту\lr1'
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']
PS D:\DIMA_Навчання\Системи Штучного Інтелекту\lr1>

```

Рис. 1.5. Кодування міток

Завдання 2: Попередня обробка нових даних.

23.	2.5	-1.6	-6.1	-2.4	-1.2	4.3	3.2	3.1	6.1	-4.4	1.4	-1.2	2.5
-----	-----	------	------	------	------	-----	-----	-----	-----	------	-----	------	-----

Результат:

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР01	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

LR_1_task_2.py > ...
1  import numpy as np
2  from sklearn import preprocessing
3
4  #23
5  input_data = np.array([[2.5, -1.6, -6.1],
6                        [-2.4, -1.2, 4.3],
7                        [3.2, 3.1, 6.1],
8                        [-4.4, 1.4, -1.2]])
9
10 data_binarized = preprocessing.Binarizer(threshold = 2.5).transform(input_data)
11 print("\n Binarized data:\n", data_binarized)
12
13 print("\nBEFORE: ")
14 print("Mean =", input_data.mean(axis=0))
15 print("Std deviation =", input_data.std(axis=0))
16
17 data_scaled = preprocessing.scale(input_data)
18 print("\nAFTER: ")
19 print("Mean = ", data_scaled.mean(axis = 0))
20 print("Std deviation =", data_scaled.std(axis = 0))
21
22 data_scaler_minmax = preprocessing.MinMaxScaler(feature_range=(0, 1))
23 data_scaled_minmax = data_scaler_minmax.fit_transform(input_data)
24 print("\nMin max scaled data: \n", data_scaled_minmax)
25
26 data_normalized_l1 = preprocessing.normalize(input_data, norm = 'l1')
27 data_normalized_l2 = preprocessing.normalize(input_data, norm = 'l2')
28 print ("\nl1 normalized data:\n", data_normalized_l1)
29 print ("\nl2 normalized data:\n", data_normalized_l2)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

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]]

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

Labels = ['green', 'red', 'black']
[-0.92228798 0.29345527 -0.25153308]]

```

Рис. 2.1. Результат виконання

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР01	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 3: Класифікація логістичною регресією або логістичний класифікатор.

Результат:

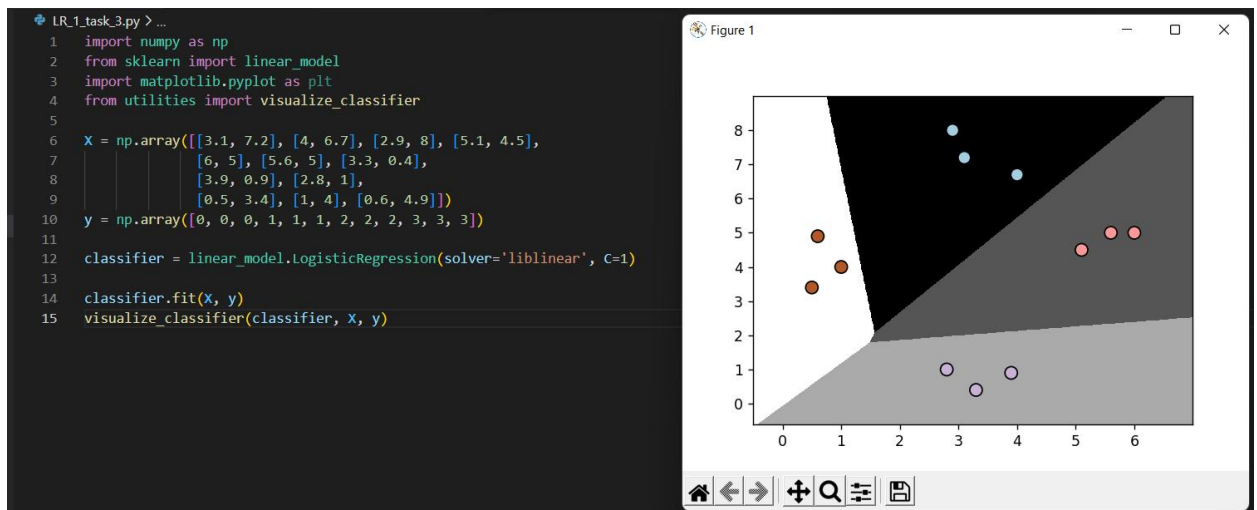


Рис. 3.1. Результат виконання

Завдання 4: Класифікація наївним байєсовським класифікатором.

Результат:

Обидва прогона дали однаковий результат, тому як генерувались однакові набори даних для навчання і тестування.

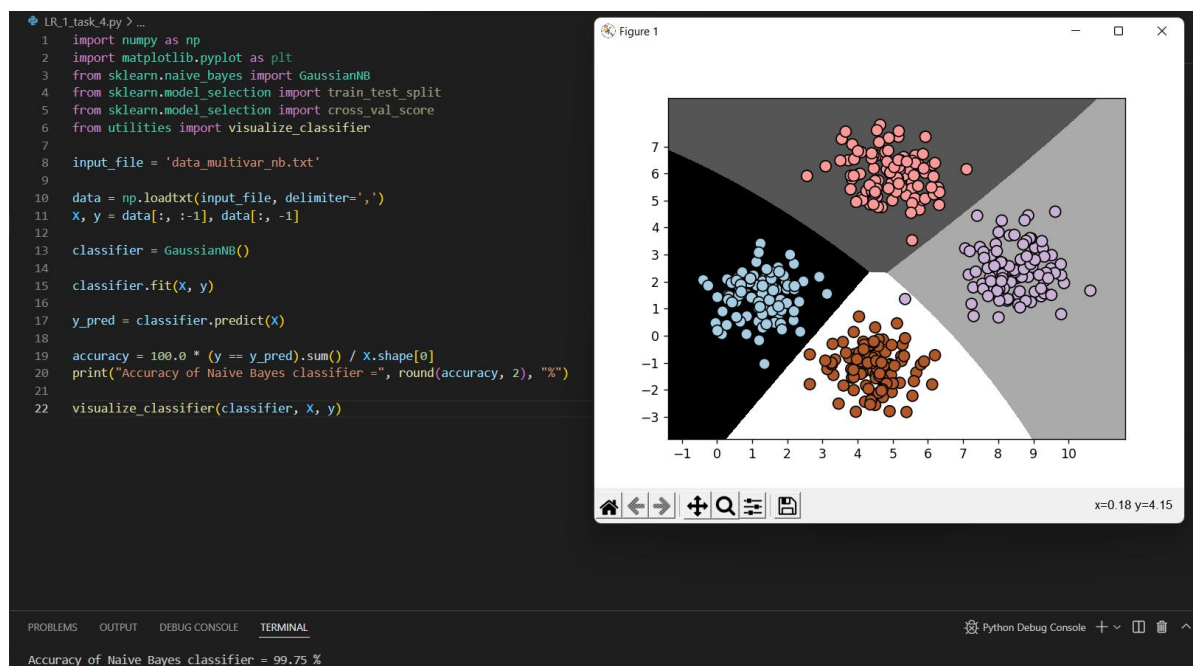


Рис. 4.1. Класифікація наївним байєсовським класифікатором

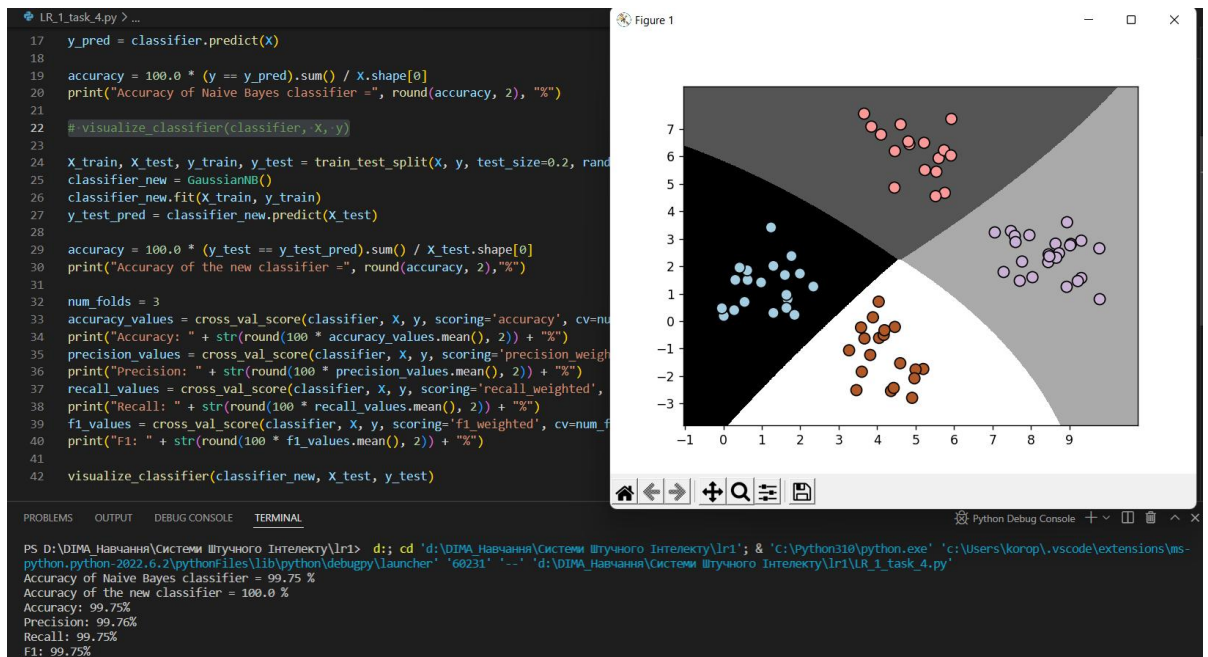


Рис. 4.1. Класифікація наївним байєсовським класифікатором з обчисленням якості точності і повноти

Завдання 5: Вивчити метрики якості класифікації.

Результат:

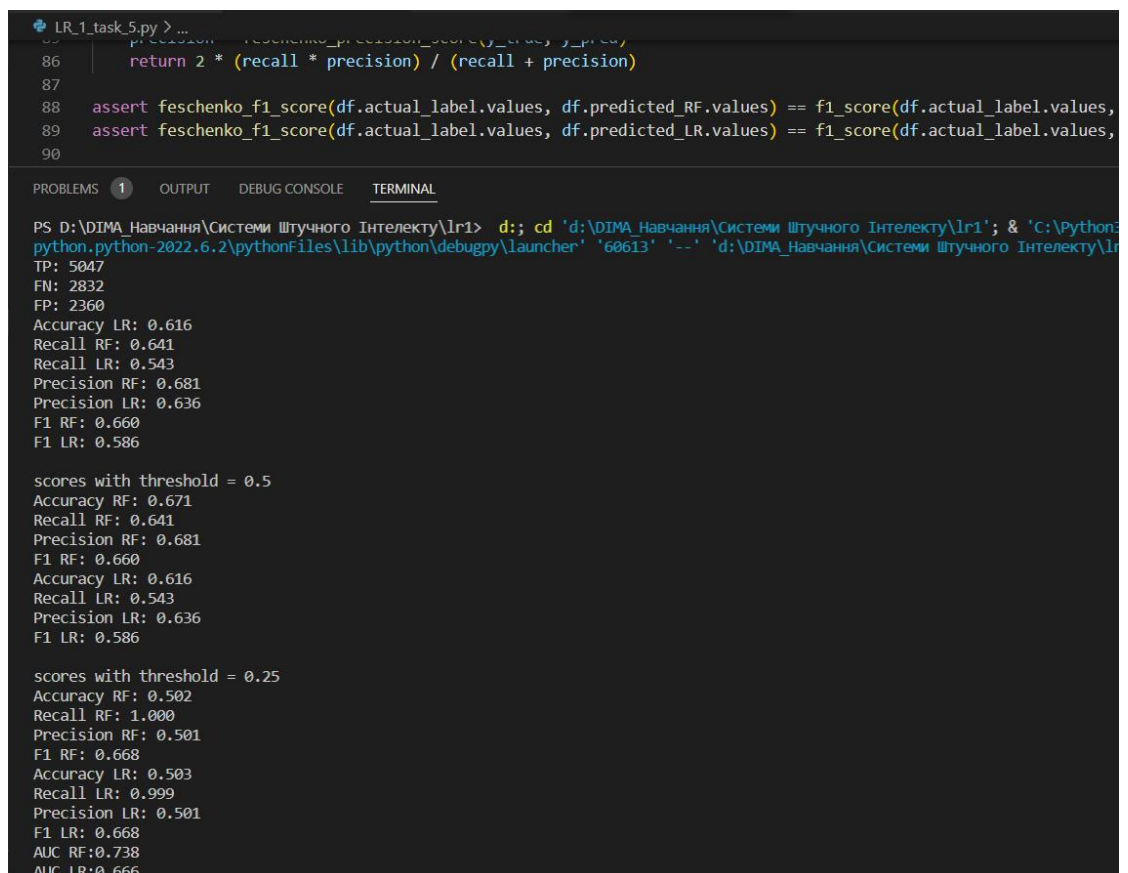


Рис. 5.1. Порівняння моделей RF та LR на кроках 0.25 і 0.5

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР01	Арк. 7
Змн.	Арк.	№ докум.	Підпис	Дата		

Порівнюючи моделі при порозі 0.25 якість та точність вищі (RF), при порозі 0.5 модель справляється краще (LF). Тому залежно від вхідних даних необхідно робити вибір.

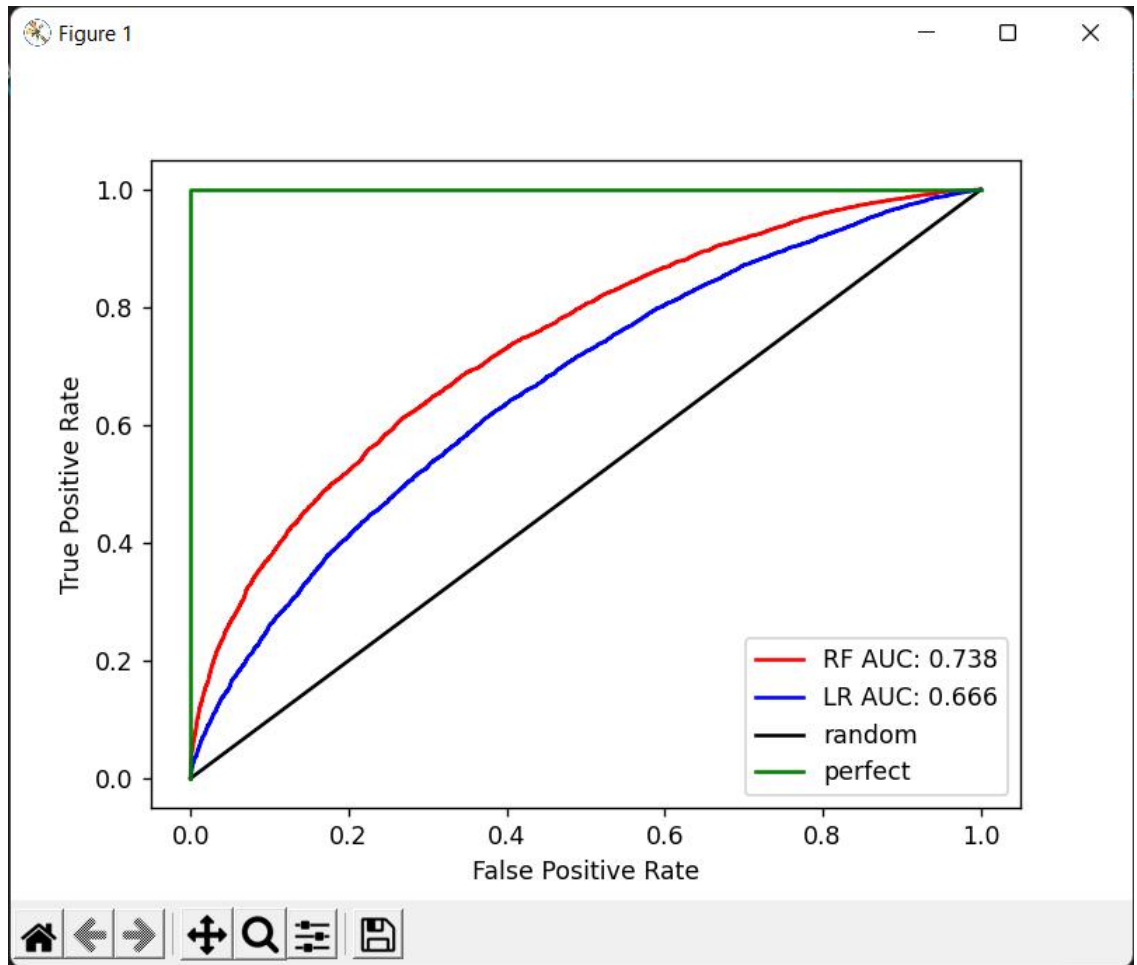


Рис. 5.2. Порівняння моделей за допомогою кривих ROC

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.

Результат:

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР01	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

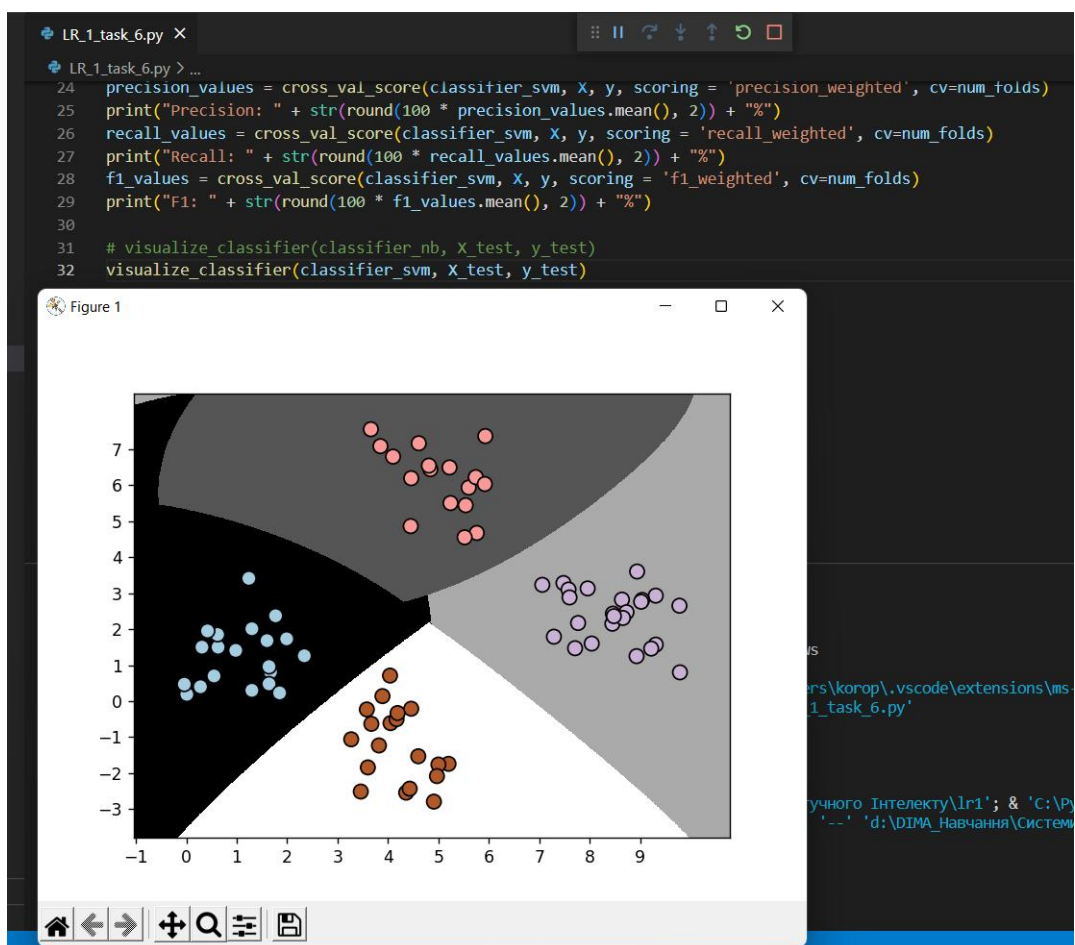
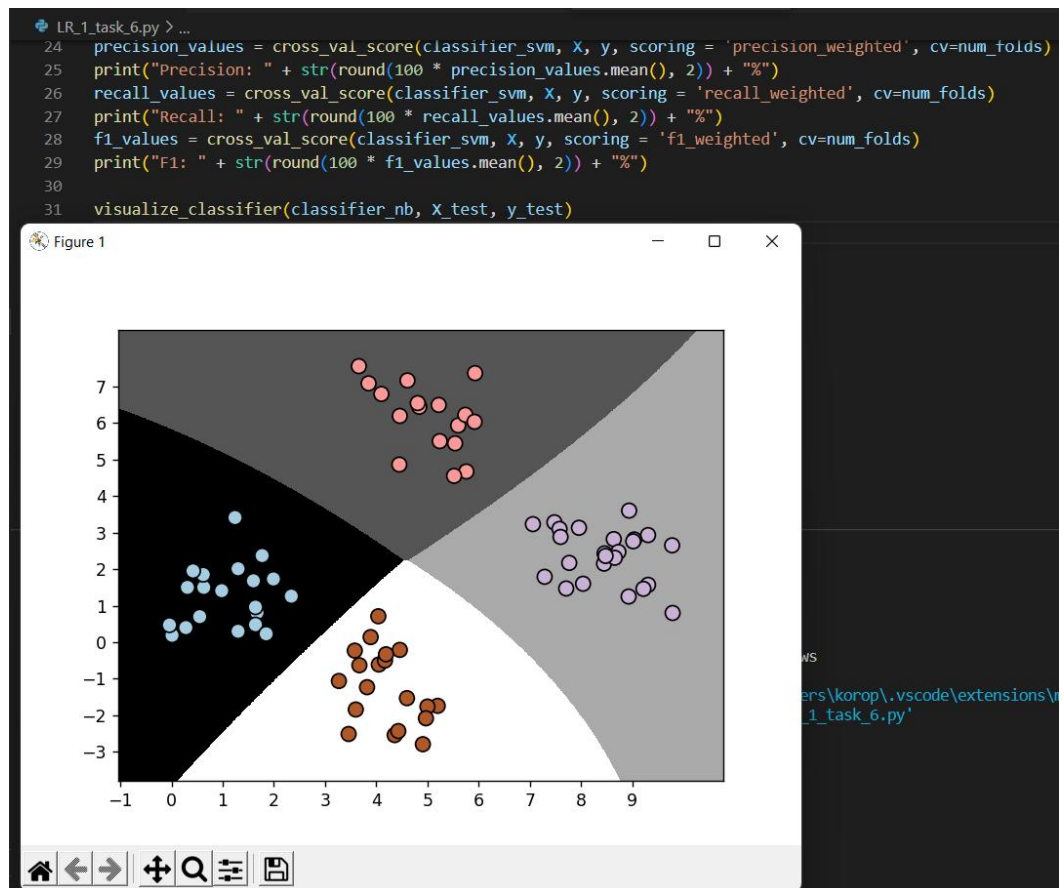


Рис. 6.1. Порівняння класифікаторів наївного байєса та SVM

Висновок: на цій лабораторній роботі ми використовуємо спеціалізовані бібліотеки та мову програмування Python дослідили попередню обробку та класифікацію даних.

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР01	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		